UNIVERSITY OF TWENTE.

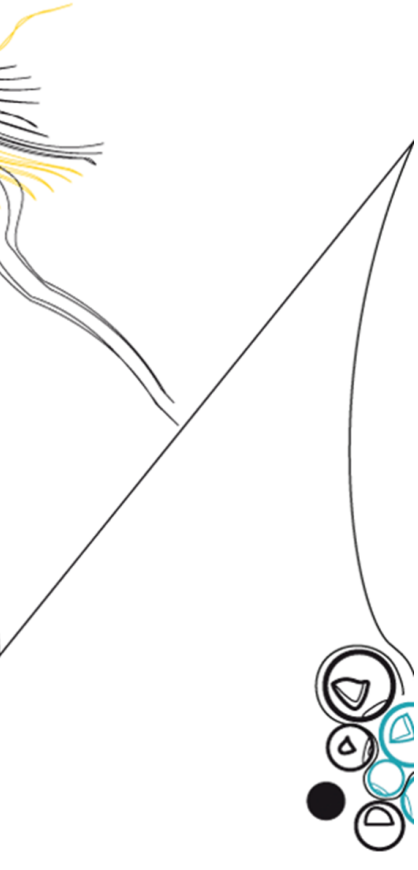
Does it make sense?

Analyzing coherence in longer fictional discourse on a syntactic and semantic level

Laura Rieke Scheuter

MSc Thesis

August 2021



Examination committee:
J. E. Van Stegeren MSc
dr. M. Theune
prof. dr. G.J. Westerhof

Human Media Interaction Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Preface

This thesis marks the end of my MSc Interaction Technology at the University of Twente, but it does even more than that: it marks the end of my study career and my student life. I therefore want to take this opportunity to look back on a period that I dare say has had a significant impact on who I have become as a person. You'll notice that the subject of this thesis logically follows from the paths I have taken to end up where I am now.

I started my study career at Utrecht University, where I started studying Language and Culture. After half a year, I had to admit that this wasn't the study for me. I missed the practical skills and the translation from theory to practice. In September of the next year, I started with the bachelor Communication Science at the University of Twente. This appeared to be more my cup of tea and I really enjoyed most of it, although I knew that I still missed some technical depth. I therefore chose to do a minor in Creative Technology and I immediately knew that this was the direction in which I wanted to continue my study career. After I finished my bachelor, I did a premaster to upgrade my programming and mathematical skills, so I would be qualified to start with the master Interaction Technology. During this master I discovered that my interests lie in data science, machine learning and specifically, natural language processing. I feel that graduating in natural language processing summarizes the quest that I call my study career. On top of that, I could use the books that have shaped my childhood (who am I kidding: they are shaping me still) as dataset: the Harry Potter books, by J.K. Rowling. This thesis truly feels like the icing on the cake.

I could not have carried out this thesis the way I did without certain people. First and foremost, I want to thank my supervisors from the University of Twente. Thank you Judith van Stegeren, for our (bi)weekly meetings and your supervision. Our meetings were insightful, keeping me on the right track, and fun. Mariet Theune and Gerben Westerhof, thank you for your valuable feedback: it really helped me taking this thesis to a higher level. Thanks to Joel Tetreault from Grammarly, for providing me with the Grammarly Corpus of Discourse Coherence. A big thank you goes out to my friends: Thieu, Wendy, Emie, Jop, Rosa, Jesper, Ymke, Chiara, Kirsten, Jorine, and everyone else: thanks for our frequent study sessions, back-and-forth rubber-ducking, and all the fun we had. You contributed significantly to making this

graduation period pretty fun indeed. And thank you to my colleagues at TRIMM, for your flexibility, encouragement, and for being a welcome distraction during this period. Lastly, I want to thank my family. Pap, Mam, Juud, Tijm: me and my ever changing mind were inimitable at times, but I am so grateful for your patience and support. I would have given up a long time ago if you didn't kick my ass every now and then.

Summary

Discourse coherence has been the topic of research for years, yet the subject seems to be untouched when it comes to longer discourse. I therefore raise the question: how can longer fictional discourse of at least 50.000 words be assessed in terms of coherence?

To analyze discourse coherence, I first distinguish between syntactic coherence (concerning grammar) and semantic coherence (concerning the meaning of the text). Using Barzilay and Lapata's Entity Grid model for the assessment of syntactic coherence, a feature vector per document can be created representing the probabilities of the sentence-to-sentence transitions of the syntactic roles of the entities present in the document. For the assessment of semantic coherence, using Global Vectors (GloVe), a vector representation of a document is created, after which the semantic similarity of adjacent sentences could be computed using the cosine similarity score. Both the feature vectors and the cosine similarity scores then are the input to two models: a logistic regression model and a random forest model. Coherence is evaluated in three ways: using only the feature vectors, using only the similarity scores and finally using both the feature vectors and the similarity scores.

The coherence analysis is applied to three datasets, starting with the Grammarly Corpus of Discourse Coherence (GCDC). This dataset is rated in terms of coherence on a three-point scale but only consists of approximately 9 sentences per document. I manipulated the documents in this dataset to create a second dataset: the GCDC two-point scale dataset, containing only two coherence classes. This was done because the third dataset, the books dataset, also contained only two coherence classes. Generated books are used as a ground truth for low coherent discourse, and the Harry Potter books by J.K. Rowling are used as a ground truth for highly coherent discourse. All books are first divided in snippets of approximately 9 sentences, to resemble the GCDC data.

Finally, the method is expanded to longer snippets (newly created 'chapters') of the generated books and the Harry Potter books. These chapters contain 10 snippets and thus, consist of approximately 90 sentences. The prediction of the models on the chapters is used to predict whether the books are either low coherent or highly coherent: if most chapters were predicted to be low coherent, the book is

predicted to be low coherent; if most chapters of the book are predicted to be highly coherent, the book is predicted to be highly coherent.

The results show that the models did not perform well on the syntactic and semantic coherence features of the GCDC original data. They did perform well on the GCDC two-point scale data and the books data. This indicates that the proposed methodology is suitable for distinguishing handwritten discourse from generated discourse, but less suitable for distinguishing highly coherent discourse from low coherent discourse. The performance of the models increased when provided with the syntactic and semantic coherence features of the chapters, which indicates that the proposed methodology is fitting for longer discourse.

One of the biggest limitations was to acquire data that was rated in terms of coherence, especially when it came to longer discourse. I would therefore recommend future researchers to focus on creating such a dataset. Also, as the performance of the models increased as the length of the discourse increased, it could be worthwhile to focus on developing and improving efficient language models.

Contents

Acronyms

CBOW	Continuous bag-of-words
CT	Centering Theory
DT	Decision tree
GCDC	Grammarly Corpus of Discourse Coherence
GloVe	Global Vectors
LDA	Latent Dirichlet Allocation
LR	Logistic regression
LSA	Latent Semantic Analysis
MCS	Mean cosine similarity
MTurk	Amazon Mechanical Turk
NaNoGenMo	National Novel Generation Month
NaNoWriMo	National Novel Writing Month
NER	Named Entity Recognition
NLG	Natural Language Generation
NLP	Natural Language Processing
OOB	Out of Bag
OOV	Out of vocabulary
POS	Part-of-speech
RF	Random forest
RST	Rhetorical Structure Theory
Std	Standard deviation
SVD	Singular Value Decomposition
SVM	Support Vector Machine

Chapter 1

Introduction

In 1999, freelance writer Chris Baty came up with the idea to start a writing challenge: write a novel of at least 50.000 words in only one month time. 21 participants joined the challenge, and so, in that year, National Novel Writing Month (NaNoWriMo) found its origin. Over 20 years have passed, and NaNoWriMo has become a non-profit organisation, putting up the NaNoWriMo challenge with hundreds of thousands of people accepting the challenge to write a novel during the month of November annually.

Yet as with so many things, times are changing, and so, after a few years, a new challenge originated. Inspired by NaNoWriMo, computer programmer Darius Kazemi initiated the National Novel Generation Month (NaNoGenMo) in 2013. The idea is quite similar but instead of writing a novel manually, contestants are challenged to write code that generates a novel. The results are varied: as the only requirement to take part in the challenge is to generate a novel of at least 50.000 words, some submissions lack any form of story line, whereas other submissions resemble a handwritten novel more accurately. The definition of 'novel' in this challenge is very vague. As the organization of NaNoGenMo states on their website: *The "novel" is defined however you want. It could be 50,000 repetitions of the word "meow". It could literally grab a random novel from Project Gutenberg. It doesn't matter, as long as it's 50k+ words.*[1]

In order to be able to meaningfully compare the submissions, one would need to read all novels and judge them afterwards: it would be a time consuming task and also, it would be prone to subjectivity. A novel can be 'good' or 'bad' in many ways, and it is therefore important to define the criteria on which the quality of the novel is judged. An important criterion for assessing text quality is coherence - the degree to which a text forms a whole, either syntactically (concerning grammar) or semantically (concerning the meaning of text). Much research has been conducted on the topic of coherence, yet the subject seems to be untouched when it comes to

[1] https://nanogenmo.github.io/

longer discourse. This raises an even bigger question that goes beyond the purpose of assessing the NaNoGenMo results: how can fictional discourse be automatically assessed in terms of coherence?

1.1 Research questions

The main research question has been formulated as follows:

> How can a fictional discourse of at least 50.000 words be automatically assessed in terms of coherence?

This research question will be answered using the following sub-questions:

> What is the added value of resolving coreferences when analyzing coherence?
>
> How can a document of approximately 9 sentences be automatically assessed in terms of syntactic coherence?
>
> How can a document of approximately 9 sentences be automatically assessed in terms of semantic coherence?
>
> How can the analysis of syntactic and semantic coherence be combined?
>
> How can the analysis of syntactic and semantic coherence in short documents be extended towards fictional discourse of at least 50.000 words?

1.2 Thesis overview

Chapter 2 starts with providing an overview of existing literature in the field of discourse coherence. This chapter, like this introduction chapter, is largely based on the preparatory research that I conducted in the Research Topics course. In Chapter 2, I explain the difference between syntactic and semantic coherence, and I show some existing computational approaches to coherence. The concept of coreference resolution is also discussed here, and why this could probably be important when analyzing coherence. Then, in Chapter 3, I show, analyze and prepare the three datasets that were central to this research. Thereafter, in Chapter 4, I describe how I applied coreference resolution to the documents in all datasets using an algorithm. I show and analyze the results of this algorithm. The next chapters, Chapter 5 up until Chapter 7, describe how I analyzed coherence in documents of approximately 9 sentences. I start with analyzing syntactic coherence in Chapter 5, after which I

analyze semantic coherence in Chapter 6, and then, in Chapter 7, I synthesize the results from the syntactic and semantic coherence experiment. Finally, in Chapter 8, I describe how I extended the methodology that I previously applied on shorter documents towards longer discourse. In each chapter, I discuss the results of the specific experiments and draw some partial conclusions. However, in Chapter 9, I present a general discussion that addresses some overarching results and recommendations. Lastly, in Chapter 10, I wrap up this thesis by answering the research questions that were presented in Section 1.1.

Chapter 2

Background

Coherence is an important concept when studying the process of discourse comprehension. It has been explored extensively, though various scholars seem to use slightly different definitions. Back in 1993, Spencer and Fitzgerald [32] already found 27 distinct definitions of coherence. They also found that often, coherence is interchangeably used with cohesion, yet Morris and Hirst [27, p. 25] stress the difference between these terms: "[...] cohesion is a term for sticking together; it means that the text all hangs together. Coherence is a term for making sense; it means that there is sense in the text". This means that a discourse can contain cohesive text, although the discourse is incoherent, and the other way round.

Mann and Thompson describe a fully coherent discourse as a discourse that only consists of segments that logically follow each other [21]. Mann states: "every part has some function [...] and furthermore, there is no sense that some parts are somehow missing" [20]. For a story to be coherent, all the parts of the story must be structured so that the entire sequence of events is interrelated in a meaningful way [13].

Grosz and Sidner presented a theory of discourse structure in which they identified three components of discourse structure: the linguistic structure, the intentional structure and the attentional state [9]. The linguistic structure describes the relationship between different discourse segments, which are aggregations of utterances in a discourse. The intentional structure represents the purpose (intention) of a discourse segment and the relationship between the purposes of the different discourse segments. Finally, the attentional state refers to the focus at a certain point in the discourse of the discourse participants and depends on both the linguistic and the intentional structure of the discourse. The intentional state can be analyzed within a segment, but also over the entire discourse.

The concept of coherence is very complex, but becomes more easy to define when distinguishing between local coherence and global coherence. Local coherence describes the relationship within a discourse segment, between textual ele-

ments on the level of sentence-to-sentence transitions [1]. It neglects to take into account the overall coherence of a discourse, or coherence between the different discourse segments. This is where global coherence comes in. Global coherence is about a certain kind of structure that can be found throughout the discourse. According to Grosz and Sidner, each of the aforementioned components of discourse structure also have a local and a global level [9].

In addition to this, we can distinguish between syntactic coherence and semantic coherence. The first one, syntactic coherence, is the most similar to the earlier mentioned definition of coherence by Mann and Thompson [21] and comprises a more technical approach to coherence. I will elaborate on this concept further in Section 2.1. The second one, semantic coherence, tells something about the topic of the discourse at hand. A discourse that is coherent semantically contains words that are from the same or adjacent semantic fields. In Section 2.2, I will elaborate more about this topic.

2.1 Syntactic coherence

Syntactic coherence describes the grammatical structure in the text. This approach doesn't take into account the meaning of the words that are used in the discourse, but rather analyzes the way the discourse is structured grammatically. A story can be syntactically coherent, but at the same time be complete nonsense in terms of its meaning. Example 2.1 illustrates this.

Example 2.1
John walks his dog. He buys a rose while he is cycling, but the dog waves his wand. John sees an airplane crash but he is not flying because his dog drinks a cup of gasoline.

This short piece of text is hard to interpret for a human reader, since the words that are used have nothing to do with each other. However, the text is grammatically correct, which means that the text would score high in terms of syntactic coherence.

2.1.1 Coherence relations

Coherence relations can be used to analyze syntactic coherence. They describe the way that two text segments, often clauses or subordinate clauses, are (rhetorically) related to each other [21] [30]. The nature of this relationship can be, for example, reason, elaboration, or evidence [5]. The relationship can be made explicitly distinguishable by the use of cue phrases: signal words or signal sentences that

announce a coherence relation [9] [15]. However, cue phrases are not necessarily required for a coherence relation to exist. Discourse segments can also share a coherence relation when cue phrases are absent.

A popular theory that models coherence relations is Rhetorical Structure Theory (RST) by Mann and Thompson [21]. In RST, a text span is defined as an uninterrupted linear interval of text. Each text span can be a nucleus or a satellite, a pair of text spans contains a nuclear and a satellite and these two spans have an asymmetric relationship with each other. A nucleus is the text span that is prominent for the writer of the text; a satellite is less prominent and depends on the nucleus regarding interpretability. Sometimes a symmetric relationship between two text spans occurs: in these cases, both text spans are nuclei.

2.1.2 Entity-based coherence

Entity-based coherence is a more psychological approach to coherence. The main idea is that the more a discourse is about the same entity or entities, the more coherent the discourse is. This is not necessarily indicated by using the same word for one entity throughout the discourse; it is also possible that different words refer to the same entity by the use of referring expressions. This is called coreferencing [14] [31]. The entity that is referred to is called the referent, whereas the words that are used to address this entity are called the referring expressions. Models designed to assess entity-based coherence should address coreference resolution in order to gain optimal results. After all: if a model treats different references to the same entity as totally different entities, the results yielded by the model will deviate significantly from the truth. The discourse segment in Example 2.2 illustrates this.

Example 2.2
John walks his dog. He meets Charlie. She is his girlfriend.

As humans, we are able to deduce that ‘John’ (referent) and ‘he’ (referring expression) refer to the same entity, and so do ‘Charlie’ (referent), ‘she’ (referring expression), and ‘his girlfriend’ (referring expression). If a model would treat these occurrences as different entities, it would lower the coherence score of the discourse. It is conceivable that this effect will increase as the discourse under consideration gets longer.

Centering Theory

Centering Theory (CT) can be used to analyze entity-based coherence [10]. CT is based on the idea that at any moment in a discourse, at least one of the entities

is salient ('centered on' or 'in focus') and that coherence is determined by the unity between the salient entities and the words that refer to these entities. In CT, a discourse segment contains utterances and each utterance U is assigned a set of forward-looking centers: $C_f(U)$. Each utterance U in the discourse segment after the first utterance is assigned one backward-looking center: $C_b(U)$. The idea of CT is that each backward-looking center of U_{n+1} has some relation with the forward looking center of U_n. Furthermore, three types of transition relations across pairs of utterances are defined within CT: center continuation, center retaining, and center shifting. These centering transitions help to determine the coherence of a discourse segment.

To illustrate CT, consider Example 2.3 from the original paper [10, p. 7]:

Example 2.3

1. (a) John went to his favorite music store to buy a piano.
   (b) He had frequented the store for many years.
   (c) He was excited that he could finally buy a piano.
   (d) He arrived just as the store was closing for the day.
2. (a) John went to his favorite music store to buy a piano.
   (b) It was a store John had frequented for many years.
   (c) He was excited that he could finally buy a piano.
   (d) It was closing just as John arrived.

In Example 2.3 (1), the salient entity stays the same throughout all sentences: the entity 'John' (later referred to as 'he') is salient in all four sentences. In Example 2.3 (2) however, the salient entity shifts from John, to the store, back to John, and to the store again. According to CT, this second example is therefore less coherent than the first one.

Entity Grid Model

The Entity Grid Model [1] assumes that each discourse can be represented in a matrix, in which each row represents a sentence and each column represents an entity, which is defined as "a class of coreferent noun phrases". The result is an entity grid in which each occurrence of an entity has its own cell in which information about the occurrence can be stored: is the entity present or absent in the sentence? If it is present, which syntactic role does the entity have in the sentence? This information is stored in a cell using predefined labels, and can be extracted from the text using a dependency parser. Coherent texts are expected to result in an

entity grid with some dense columns with subject and object labels, meaning that the entity described in that column plays a role throughout the entire text, and more sparse columns, describing entities that are not important for the text.

To illustrate this, consider Example 2.4 and the entity grid in Table 2.1 from the original paper [1, p. 6-7].

Example 2.4

1. The Justice Department is conducting an anti-trust trial against Microsoft Corp. with evidence that the company is increasingly attempting to crush competitors.
2. Microsoft is accused of trying to forcefully buy into markets where its own products are not competitive enough to unseat established brands.
3. The case revolves around evidence of Microsoft aggressively pressuring Netscape into merging browser software.
4. Microsoft claims its tactics are commonplace and good economically.
5. The government may file a civil suit ruling that conspiracy to curb competition through collusion is a violation of the Sherman Act.
6. Microsoft continues to show increased earnings despite the trial.

Table 2.1: A fragment of the entity grid. Noun phrases are represented by their head nouns. Grid cells correspond to grammatical roles: subjects (S), objects (O), or neither (X).

	Department	Trial	Microsoft	Evidence	Competitors	Markets	Products	Brands	Case	Netscape	Software	Tactics	Government	Suit	Earrings	
1	S	O	S	X	O	-	-	-	-	-	-	-	-	-	-	1
2	-	-	O	-	-	X	S	O	-	-	-	-	-	-	-	2
3	-	-	S	O	-	-	-	-	S	O	O	-	-	-	-	3
4	-	-	S	-	-	-	-	-	-	-	-	S	-	-	-	4
5	-	-	-	-	-	-	-	-	-	-	-	-	S	O	-	5
6	-	X	S	-	-	-	-	-	-	-	-	-	-	-	O	6

The entity grid in Table 2.1 contains the head nouns of all entities from Example 2.4 as column headers. The rows represent the sentences from the example. Each cell reflect whether the entity is absent or present in the sentence, and if the entity is present, it reflects the grammatical role of the entity in the sentence.

The analysis of this entity grid is based upon CT. The local entity transitions (sentence-to-sentence transitions) are visible in the grid, and each transition comes with a certain probability p (computed by the amount of times the transition of length *n* occurs throughout the discourse, divided by the total amount of transitions of length *n* in the discourse). This means that the discourse at hand can be regarded as a distribution of transition types, which can be represented as a feature vector:

$$\Phi(x_{ij}) = (p_1(x_{ij}), p_2(x_{ij}), ..., p_m(x_{ij}))$$

In this feature vector, x_{ij} represents the entity grid, m is the total number of distinct entity transitions, and $p_t(x_{ij})$ is the probability of a specific transition type in the entity grid.

Barzilay and Lapata do not explicitly propose a scoring function for the assessment of entity based coherence based upon the entity grid. They argue in favor of a ranking system, in which the ranking is learned using a Support Vector Machine (SVM). To evaluate their model, they make permutations to their original data and subsequently, they use a pairwise ranking: the model should prefer the original text above the permutations and thus, rank it higher.

2.2 Semantic coherence

Thus far, I have discussed syntactic approaches to measure text coherence. However, as stated before, the semantic meaning of the discourse is completely disregarded in these approaches. It could therefore be a valuable addition to also investigate more semantic approaches to coherence. In the previous paragraph I have shown an example of a syntactically coherent, but semantically incoherent text. Example 2.5 shows that the other way round is also true: a story can contain words from the same semantic field, but at the same time, be syntactically incoherent.

Example 2.5
John dog walking line. Poop barking furry friend. Tail happy wagging woof woof barking.

Semantic coherence has been the subject of research before. Halliday and Hasan [11] first introduced the concept of lexical cohesion. They state that lexical cohesion is “the cohesive effect achieved by the selection of vocabulary”. Words that share a semantic meaning, or in other words, words that are semantically cohesive, indicate semantic coherence. Words that occur together in a discourse segment and that are from the same semantic field, will therefore increase the semantic coherence of

that discourse segment [12]. Lexical cohesion is explicitly differentiated from grammatical cohesion: the latter indicates grammatical, or syntactic coherence, whereas the first indicates semantic coherence [29].

2.2.1 Topic Modelling

Topic modelling is a technique to automatically learn latent topics from one or more documents. Latent topics are features that are implicit to the documents, but cannot be directly measured. In practice, this technique is often applied to learn topics from journals or articles. This way, clusters of similar articles can be detected, which facilitates systematic searching for specific information. In this context, topic coherence has been the subject of research to understand how and to which degree different latent topics cohere and also, to which degree the documents within a topic cohere.

Latent Dirichlet Allocation

A popular model for automatically learning topics from a set of documents is Latent Dirichlet Allocation (LDA) [2]. This is a generative probabilistic Bayes model implementation. LDA is based upon the assumption that each document contains latent topics, meaning that these topics are not necessarily explicitly visible within these documents. The topics are represented by a distribution over non-latent words. When these words are retrieved from the documents, they can subsequently be used to derive the latent topics. Research shows that LDA is a flexible and strong model in terms of topic coherence [33].

To assess the quality of the latent topics that are extracted, word similarity metrics are used on the clusters of non-latent words. These metrics determine to what extent the non-latent words are similar: the higher the similarity score, the more similar the words are, and thus, the better the latent topics are. An example of such metric is the Umass score [25]. This score was designed to optimize semantic coherence within topics. This score is based upon the idea that a topic is a list of a predefined number of most probable words within the topic: $V^{(t)} = (v_1^{(t)}, ..., v_m^{(t)})$. The metric assesses the semantic coherence of a topic based on the document frequency of word v, and the co-document frequency of word v and v'.

Latent Semantic Analysis

Another method for topic modelling is Latent Semantic Analysis (LSA). This algorithm can be used to determine the semantic relatedness of two discourse segments [8]. A discourse segment can consist of a sentence, paragraph, document, or even an entire collection of documents: for this reason, LSA is also often used to extract topics within a collection of documents and find the semantic relatedness of the documents by clustering them into the extracted latent topics.

With LSA, a discourse segment is represented in a so-called document-term matrix. In this matrix, each column is a discourse segment and each row is a unique word. The cells contain the word counts per discourse segment. Since this can result in an extremely large matrix, a dimensionality reduction technique called Singular Value Decomposition (SVD) is applied. With SVD, the number of rows is reduced but the structure along the columns is preserved: it reduces the matrix to its constituent form.

The resulting matrix columns can be regarded as a feature vector. Each vector has a certain location in a high dimensional semantic space. LSA is based upon the distance between two vectors of adjacent discourse segments. This distance is computed using a cosine, and it is called the cosine similarity: the smaller the cosine, the bigger the distance between the vectors [7]. The cosine similarity score of a discourse can be computed by taking the mean of the cosines of the adjacent discourse segments within the discourse [8]. Discourse with a small cosine similarity score is considered incoherent, whereas coherent discourse will return a bigger cosine similarity score.

2.2.2 Word embeddings

Word embeddings are numerical representations for words. This numerical representation, a vector, approximates the lexical meaning of the corresponding word. Each word, or token, is represented as an n-dimensional vector, and thus, has a corresponding location in an n-dimensional semantic vector space. Each dimension of the vector describes a feature of a word, but these features are not human interpretable: they are just numbers (floating points) that are obtained by a machine learning model. The theory behind this is that words with similar semantic meaning occur often in the same context. They correspond to similar vectors, and thus, are situated close to each other in the semantic space. Since it is possible to calculate with vectors, word embeddings make it possible to calculate with words and like that, find analogies. Example 2.6 is a famous illustration of what this means in practice [24].

Example 2.6
King - Man + Woman = Queen

Intuitively, the equation in Example 2.6 is correct: if you would take a king, subtract the man characteristics from a king and add woman characteristics to it, the result would be a queen. However, we cannot simply compute with words like this. Fortunately, word embeddings make it possible to do this, or rather: approximate this.

If a machine learning model would learn the vector representations of the words in Example 2.6 and we would replace the words with their corresponding vectors, the equation approaches a perfect solution. A vector for king minus the vector for man, plus the vector for woman, would approach the vector for queen. Following this, it is also possible to compute the distance in semantic meaning between two words by computing the distance between the corresponding word vectors using the cosine between the vectors, as previously explained in Section 2.2.1.

Though before we can do this, first, the numerical representations of words have to be learned. Multiple methods to learn these numerical representations exist. I will discuss two methods that were popular in recent years, namely Word2Vec and Global Vectors (GloVe).

Word2Vec

Word2Vec was patented by Google in 2013. The model can be used with a pre-trained embedding layer, which means that in that case, the model contains learned vectors which can be applied to other datasets as well. It is also possible to train the embedding layer on a new dataset.

There are two types of Word2Vec models: the continuous bag-of-words (CBOW) model [22] and the skip-gram model [23]. The most important difference between these models can be found in how these models are trained: the CBOW model aims to predict a target word given a set of context words, whereas the skip-gram model works the other way round.

With CBOW, the average of the vectors of the context word is computed. This results in a new vector, which is compared to the vector of the target word. The expectation is that the average vector approaches the vector of the target word. Therefore, the loss of the model is computed from the average vector and the target word vector. Lastly, the weights of the model are updated using back propagation.

Skip-gram works the other way round. It takes the word vector of the target word and one other (context) word. These are merged - not by taking their average, but by taking their dot product. The resulting vector is passed into a sigmoid layer. This then results in a classification of either 1 (meaning the word is a true context word) or 0 (meaning the word is not a context word). Using this result and the truth value (whether the word is a true context word or not), the loss of the model is computed and the weights are updated using back propagation. However: this means that although the other word might not be a context word in the example under consideration, it could be a context word for another occurrence of the target word in the text.

This illustrates one disadvantage of these models - models that work with a sliding local context window. It means that only the current target word's neighbouring words are taken into consideration when learning the word similarities. The advan-

tage of Word2Vec over more traditional methods (like LSA) is that both the CBOW model and the skip-gram model are able to find analogies. However, an important disadvantage is that because of the local nature of these methods, they do not take the global statistics of the data into account, hereby missing important information and being unable to handle out of vocabulary (OOV) words.

GloVe

These disadvantages are addressed with Global Vectors (GloVe), a word embeddings model that has gained popularity in recent years. GloVe is also a pre-trained method, which not only takes the local context window into account when learning the word vectors. On top of that, it utilizes the global statistics of the dataset using the idea of a co-occurrence matrix, an idea that for example LSA was also built upon [28]. The co-occurrence matrix is basically a count matrix. Both the rows and the columns represent the words that occur in the dataset. The cells reflect how many times the corresponding words occur in each others context in the dataset. The size of the context is a parameter that can be changed: for example, if the context has size 1, only a word's direct neighbours are considered as its context.

From this co-occurrence matrix, the probability P that a word j appears in another word's i context, or the probability that a word i appears *given* another word k, $P(i|k)$, can be computed as follows:

$$P_{ik} = P(i|k) = X_{ik}/X_i$$

In the above equation, X represents the co-occurrence matrix, X_{ik} represents the amount of times word i appears in the context of word k and X_i represents the amount of times word i appears in the dataset (in the context of any other word). Since these probabilities are not easily interpretable for large datasets, GloVe is not based solely on these probabilities, but on the ratio of the probabilities of two words i and j given the same context word k: the co-occurrence probabilities. To illustrate this, consider Table 2.2 from the original paper [28].

Table 2.2: Co-occurrence probabilities illustrated, from [28]

Probability and Ratio	*k = solid*	*k = gas*	*k = water*	*k = fashion*
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Table 2.2 illustrates that the ratio (co-occurrence probability) of the probabilities of two target words i and j given the same context word k contains information about

whether that context word k is related to the target words i and/or j. If the co-occurrence probability results in (approximately) 1, the context word is either related to both words or to neither. In Table 2.2, this is the case for context words $k\ water$ and $fashion$: $water$ is related to both ice and $steam$, whereas $fashion$ is related to neither. The co-occurrence probability is computed by dividing the probability that word k occurs *given* that word ice occurs ($P(fashion|ice)$ or $P(water|ice)$), by the probability that word k occurs *given* that word $steam$ occurs ($P(fashion|steam)$ or $P(water|steam)$), as shown in the bottom row of the column. Since these probabilities are about the same, the co-occurrence probability results in approximately 1.

If the co-occurrence probability is larger than 1, the context word is related to the first target word. The table shows that this is the case with the context word $k\ solid$: it is related to ice, but it is not related to $steam$. It will therefore co-occur with ice more than it will co-occur with $steam$. This means that the numerator of the division to compute the co-occurrence probability is larger than the denominator, and therefore, the co-occurrence probability is larger than 1.

Finally, if the co-occurrence probability is (significantly) smaller than 1, the context word is related to the second target word. This is the case with context word $k\ gas$: it co-occurs with $steam$ more than it co-occurs with ice. The denominator of the division to compute the co-occurrence probabilities is larger than the numerator, and the co-occurrence probability is smaller than 1.

This is a big difference between this global technique and the local technique that is used with Word2Vec. Here, all the context words of a target word in the entire dataset are used, whereas with Word2Vec, only the context words of a specific occurrence of a target word are considered.

However, if we would use all the words in the corpus, it would result in an extremely large matrix and the benefits of word embeddings would not be utilized. This is where the word vectors come into place. Instead of computing the ratio in the aforementioned way, the same information is retrieved using the differences between the word vectors w_i and w_j corresponding to target words i and j and subsequently taking the dot product of this resulting vector with the word vector of the context word k, $\tilde{w}_k$:

$$(w_i - w_j)^T \tilde{w}_k = \frac{P_{ik}}{P_{jk}}$$

The vector space makes it possible to retrieve the information this way rather than by creating the complete co-occurrence matrix and computing the co-occurrence probabilities. In this equation, w_i and w_j are the word vectors corresponding to the target word and $\tilde{w}_k$ is the word vector of an arbitrary context word. This is not the final equation that underlies the GloVe model, but it does illustrate the main

differences between GloVe, Word2Vec and LSA. The complete rationale behind the GloVe model can be found in the original paper, [28].

Combining both the advantages of Word2Vec and LSA, GloVe has proven to be a very powerful method to learn word vectors and find relations and similarities between words from large amounts of text.

2.3 Inference making

The aforementioned approaches to coherence are (computational) linguistic approaches, and they treat coherence as a characteristic of the discourse at hand. This might have given the impression that coherence is a concept that is explicitly present in a discourse, and thus, can be directly derived from that discourse. However, psychological oriented scholars argue that coherence is not merely a characteristic of the text, but rather a combination of a text characteristic and the ability of the reader to fill in gaps with information that is not explicitly mentioned [3] [17]. Most texts are not explicitly coherent, they require prior knowledge of the reader in order to become coherent. This activity is called inference making: the text requires a reader to infer an implicit meaning using prior knowledge in order to become coherent [6]. Although this psychological aspect of coherence is not within the scope of this research, it is important to remember that following this, coherence can never be entirely deduced from discourse alone.

2.4 Conclusion

Many scholars have analyzed the concept of discourse coherence. Coherence seems to be hardly a characteristic of the discourse alone, but rather a combination of interpretation by the reader and a characteristic of the discourse. This makes analyzing coherence computationally challenging. Nevertheless, many computational approaches to analyzing coherence exist. They can be globally mapped into approaches to analyze coherence either syntactically (concerning grammar) or semantically (concerning the meaning of the discourse). Analyzing coherence using a combination of these approaches can be beneficial.

Next to this, processing the discourse prior to analyzing it in terms of coherence can help to reveal implicit relations that a human reader can infer from the discourse, but a computational approach cannot. An example of this is the resolution of coreferences: replace reference words with the entity they refer to.

Chapter 3

Data collection and analysis

In this research, I will make use of three data sources. The first dataset that I will use is rated in terms of coherence, provided by Grammarly [16]. I will discuss this dataset more in-depth in Section 3.1.

Additionally, I will use handwritten fictional books: the Harry Potter series, containing 7 books, written by J.K. Rowling. I will also use the National Novel Generation Month (NaNoGenMo) 2018 submissions that have an actual narrative in their output. According to [34], a total of 14 submissions contain a narrative, but since I want to use as much generated books as I use handwritten books, I will only use 7 of the submissions. The handwritten books are not rated in terms of coherence, and neither are the NaNoGenMo 2018 submissions. However, since the handwritten books are written by a professional human writer, these will be considered highly coherent. The NaNoGenMo submissions on the other hand are generated by a system, and they are considered low coherent. I will discuss this books dataset further in Section 3.2.

3.1 Grammarly Corpus of Discourse Coherence

For the purpose of this research, I requested access to the Grammarly Corpus of Discourse Coherence (GCDC) [16][1]. This corpus consists of four datasets coming from four different domains, namely 'Clinton', 'Enron', 'Yahoo', and 'Yelp'. The Clinton dataset contains the e-mails that the US state department released from Hilary Clinton's office[2], The Enron dataset contains e-mails from Enron personnel[3], the Yahoo dataset contains responses to questions that were asked on the Yahoo forum[4], and lastly, the Yelp dataset contains reviews from Yelp, a platform that offers

[1] https://github.com/aylai/GCDC-corpus

[2] https://foia.state.gov/Search/Results.aspx?collection=Clinton_Email

[3] https://www.cs.cmu.edu/ ./enron/

[4] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l

consumers the opportunity to review public services [5]. Each dataset has been subdivided in a train and test set. The train sets contain 1000 documents each and the test sets contain 200 documents each.

All documents have been rated by three expert raters and five untrained raters that took the job to rate the data in terms of coherence via Amazon Mechanical Turk (MTurk)[6]: an online platform on which certain tasks or jobs can be easily outsourced to people all over the world. All the raters were asked by the researchers to rate the texts in terms of coherence on a scale from 1 to 3, in which 1 resembles a low coherent text, and 3 resembles a highly coherent text. Both the expert raters and the MTurk raters were provided with a high-level description of the concept 'coherence'. [16]

The datasets all contain the ratings of each of the individual raters, but they also contain an 'expert label' and an 'MTurk label': this label represents the consensus between the expert raters and the MTurk raters respectively. For the purpose of this research I will only use the 'expert label', which means that I will use the consensus between the expert raters as groundtruth for the coherence level in the GCDC data.

3.1.1 Analysis

The datasets contain 13 columns: a unique ID (`text_id`), a subject (`subject`) and the actual text (`text`), the expert ratings (`ratingA1`, `ratingA2`, `ratingA3`), the MTurk ratings (`ratingM1`, `ratingM2`, `ratingM3`, `ratingM4`, `ratingM5`) and the consensus of both expert (`labelA`) and MTurk raters (`labelM`). The only deviation in this is the Yahoo dataset, which contains contains two columns instead of the '`subject`' column: '`question_title`' and '`question`'.

In the original paper, `subject`, `question_title` and `question` were only used to provide context for the annotators: they were left out of all other analyses. In my

[5] https://www.yelp.com/dataset
[6] https://www.mturk.com

Table 3.1: Basic statistics GCDC dataset per domain

Dataset	Words		Sentences	
	Mean	Std	Mean	Std
Clinton	163.0	53.6	8.9	3.4
Enron	165.3	55.7	9.2	3.6
Yahoo	138.5	38.3	7.8	3.6
Yelp	156.2	46.4	10.3	3.7
Total	155.8	50.1	9.1	3.7

research, I will therefore only take into account two columns: `text`, this will be the input to my analysis, and `labelA`, which will be the coherence label (target) of the documents under consideration. Furthermore, since the different domains (Clinton, Enron, Yahoo, Yelp) are not relevant for the purpose of this research, I will not treat the datasets as separate datasets but combine them into one big dataset instead, preserving the already existing train and test split. After inspection of the data, I found that the train and test sets both contained some documents that only consist of 1 sentence. That means these documents are not suitable for the experiments to be conducted in this research: I will be investigating the transitions of the syntactic roles of entities and the semantic similarity between sentences (as I will explain in Chapter 5 & Chapter 6). Therefore, I have removed these documents from the dataset. Table 3.1 shows some basic statistics (the average word count and average sentence count per document, after removal of documents containing only one sentence), first for each dataset separately, and finally for the datasets combined.

Analysis of labels

As stated before, the GCDC dataset is rated in terms of coherence using 3 labels, 1, 2, and 3, in which 1 is used for low coherent documents and 3 is used for highly coherent documents. Since I want to ultimately train a machine learning model on this dataset using the labels as targets, it is important to have an even distribution of

Table 3.2: Distribution of labels over documents in the GCDC dataset

(a) Original

	Label 1	**Label 2**	**Label 3**
Train	1309	792	1899
Test	245	171	384
Total	1554	963	2283

(b) After removing documents with only 1 sentence

	Label 1	**Label 2**	**Label 3**
Train	1282	791	1899
Test	245	171	384
Total	1527	962	2283

(c) Undersampled

	Label 1	**Label 2**	**Label 3**
Train	791	791	791
Test	171	171	171
Total	962	962	962

the labels over the documents. If a label is overrepresented in the data, the model will overfit for that label. Table 3.2a shows the distribution of the labels over the dataset originally; Table 3.2b shows the distribution of the labels over the dataset after the removal of the documents that contained only one sentence.

Table 3.2a and Table 3.2b show that the dataset is imbalanced: label 1 and 3 are overrepresented compared to label 2 in this dataset. This class imbalance needs to be fixed to prevent the aforementioned overfitting from happening. However, with this type of data, oversampling the minority class is not an option. It would mean tweaking the existing data, this way creating more documents in the minority class synthetically. It would, however, compromise the validity of the data. The synthetically gained data is not rated by experts like the original data, and therefore, it cannot be treated the same way. It would not be valid to transfer the labels from the original data to the synthetically gained data.

For this reason, I proceeded with the undersampling of the majority classes: the data with label '1' and '3'. I randomly removed documents from these classes in both the train and test set, until the amount of data matched the amount of data with label '2'. The resulting distribution of labels over the documents is shown in Table 3.2c.

3.1.2 Creating two-point scale data

In the end, I want to be able to generalize the method I use on the GCDC data towards the data of the handwritten and generated books. Unfortunately, it was not possible to obtain a dataset with books that were rated in terms of coherence. Such a dataset does not exist at the time of writing, and it was out of the scope of this project to create one. Therefore, I have decided to treat the handwritten books as highly coherent data (since the books are written by a professional writer) and the generated books as low coherent data (since these books are not written by a professional, or even a human). This would mean, however, that the dataset with the handwritten and generated books contains only two labels (it is 'rated' on a two-point scale), whereas the GCDC data contains three labels (it is rated on a three point scale). This is why I decided to create a new two-point scale dataset out of the GCDC data.

Initially, I wanted to use the data labeled '1' as low coherent data and data labeled '3' as highly coherent data. However, after inspection of the data, it became apparent that data with label '1' from the GCDC dataset is low coherent on a different level than the generated books are low coherent. Example 3.1 illustrates this.

Example 3.1
GCDC label 1 data
Having been on the receiving end of many of these, this one was generally quite mild. Zhang first asked after my family and plans for the weekend before proceeding into his text. He also was a bit apologetic at the end. All points similar to what we got from VFM Cui. Given several factors – no mention of canceling upcoming meetings, a tone quite different from the content of the message, some small talk, no ask for an in person meeting, and no request for higher level calls - I would judge that China is trying to respond quite carefully and deliberately. Let's see if it holds. Best Kurt

Generated books data
we bends. i think this means i love you. it have myself. except what is friable is a thing i am and i am dying and you are elsewhere and i think that i hate it. please, leave we but teach we friable but keep i leave we. a thing hurts this thing it means here other in a cliff you is far you hate absorbed that we ignore. some matter hurts some flabby we means on other in some cliff me is not you pretend this matter me was absorbed to say. give me the concepts someother thing leans on and i will keep it learn it use it to say i am myself again and again. please, touch you or be i much but need me touch me.

Although the GCDC example from the label '1' data might be low coherent, it is not incorrect English, whereas the generated books data is actually incorrect English. Hence, it would not be correct to treat the GCDC data with label '1' as if it were from the same class as the generated books data.

I therefore used another method to create a new dataset. I took only the data from the GCDC dataset with label '3'. I used the original label '3' data as highly coherent data (new label: label '4'). For low coherent data, I took the sentences of all documents with label '3' and I shuffled these sentences. Then, from this 'pile of sentences', I randomly created new documents of 9 sentences, since that was the average rounded sentence count for this dataset (new label: label '0'). Table 3.3 shows the distribution of the two labels over the documents in this newly created dataset.

Table 3.3: Distribution of labels over documents in the newly created two-point scale GCDC dataset

	Label 0	Label 4
Train	1888	1899
Test	384	384
Total	2272	2283

3.2 Books dataset

The books dataset consists of 14 books, of which 7 are handwritten and 7 are generated. The 7 handwritten books are the books from the Harry Potter series, written by J.K. Rowling. The statistics of these books are shown in Table 3.4a. The 7 generated books are submissions for NaNoGenMo 2018: I have selected the submissions that contain a storyline [34] and that contain chapters, and from these submissions, I randomly selected 7 generated novels. The basic statistics of the generated books are shown in Table 3.4b.

Table 3.4: Basic statistics of the handwritten and the generated books

(a) Handwritten books

Title	# of chapters	# of sentences	# of words
Harry Potter and the Philosopher's Stone	17	5072	78953
Harry Potter and the Chamber of Secrets	18	5451	87149
Harry Potter and the Prisoner of Azkaban	22	7450	110217
Harry Potter and the Goblet of Fire	37	12141	195511
Harry Potter and the Order of the Phoenix	38	13221	261937
Harry Potter and the Half-Blood Prince	30	9304	172819
Harry Potter and the Deathly Hallows	37	11301	201327

(b) Generated books

Title	# of chapters	# of sentences	# of words
I forced an AI to watch Santa Clause Conquers the Martians	22	5862	63456
Reaching	10	4892	50364
Silk	300	7452	67019
The Restoration Of Joihibiu	105	3598	50108
The League of Extraordinarily Dull Gentlemen	23	6139	53077
Not Your Average Ultra	23	3151	58953
Velvet Black Sky	100	3284	61076

When comparing Table 3.4a and Table 3.4b, it is immediately apparent that although the generated books contain less words than the handwritten books, they consist of more chapters. This means that on average, a chapter from a generated book is shorter than a chapter from a handwritten book. Table 3.5 shows the average statistics per book for both the handwritten books and the generated books. These numbers make this difference even more clear.

The statistics from Table 3.5 also show that the generated books data is much more varied in terms of length of sentences and chapters than the handwritten books data. The mean and standard deviation of each characteristic of the generated books lie very close to each other, whereas for the handwritten books, the standard deviation is much smaller compared to the mean. This indicates that the individual numbers in the generated books data are far from the mean, in contrast to the individual numbers in the handwritten books data. This is indeed in line with Table 3.4.

Table 3.5: Average statistics per book, for handwritten books and generated books

Data	Chapters		Sentences per Chapter		Words per Chapter		Words per Sentence	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Handwritten	28	9	319	19	5411	754	17	2
Generated	83	103	179	173	2015	1723	13	4

3.2.1 Creating short snippets from books data

When comparing Table 3.1 to Table 3.4a and Table 3.4b, it stands out that the documents in the GCDC dataset contain significantly less sentences than the books, or even than the chapters in the books. Nevertheless, I want to be able to compare the GCDC data to the handwritten and generated books. Since it is not possible to increase the average number of sentences of the GCDC data so that it matches the number of sentences in the books, I have split the Harry Potter books into short snippets of (on average) 9 sentences, since that matches the average sentence count

Table 3.6: Distribution of labels over documents in short snippets of books dataset, before and after undersampling

(a) Original

	Label 0	Label 4
Train	2853	5155
Test	712	1291
Total	3565	6546

(b) Undersampled

	Label 0	Label 4
Train	2853	2853
Test	712	712
Total	3575	3575

of the GCDC data. This resulted in 7006 short snippets for the handwritten books. These snippets are labeled '4': highly coherent. I subsequently did the same for the NaNoGenMo books, which resulted in 3565 short snippets for the generated books. These snippets are labeled '0': low coherent. I divided the data into a train and test set. The distribution of the documents over the two labels can be seen in Table 3.6a.

From Table 3.6a I concluded that this newly created dataset was imbalanced as well. This is caused by the fact that the generated books were generally shorter than the handwritten books. I therefore again randomly undersampled the majority class, which resulted in the final distribution of labels over documents as shown in Table 3.6b.

3.2.2 Creating new chapters from books data

Ultimately, I want to be able to generalize my methodology to longer discourse. I described earlier how I selected the NaNoGenMo books that are included in my dataset for the reason that they contain chapters. However, after inspection of the data in Table 3.5, it became apparent that the chapters of the NaNoGenMo books are very short, especially compared to the chapters of the Harry Potter books. Comparing the syntactic and semantic features from the chapters of the generated books to the syntactic and semantic features from the chapters of the handwritten books would produce biased results.

I therefore decided to create new chapters from the books dataset, consisting of 10 adjacent snippets of 9 sentences (that I created in the way I described in Section 3.2.1). I decided that these snippets could overlap the original chapters of the books, because if they could not, these new chapters would not be an improvement - some chapters from the generated books data only contained one snippet. The chapters all consisted of (on average) 90 sentences. This resulted in 647 handwritten chapters and 358 generated chapters. This means that the chapters with label '4' (highly coherent) would be overrepresented. I proceeded with undersampling these chapters to match the amount of chapters from the generated books, which means I ended up with 358 handwritten chapters and 358 generated chapters. In this undersampling, I made sure that the amount of chapters that remained were divided evenly over the books. An example of a newly created handwritten chapter is included in Appendix B; an example of a newly created generated chapter is included in Appendix C.

Chapter 4

Coreference resolution

This chapter describes the first experiment that I conducted in order to ultimately analyze discourse coherence. I explain how I applied coreference resolution to the documents in all datasets. I end up with two versions of the documents: the original documents and the documents in which coreferences were resolved. For all the upcoming experiments and for all datasets, I used both the data on which I applied coreference resolution and the original data (without applying coreference resolution), so I will ultimately be able to compare the results with and without resolving coreferences.

4.1 Methodology

In Section 2.1.2, I explained the concept of coreferencing and the importance of coreference resolution, especially for entity based coherence. I therefore started by applying coreference resolution on the datasets, using the Python package `neuralcoref`[1] by HuggingFace. This package is a pipeline extension for `spaCy`, an open source software library for Natural Language Processing (NLP) in Python. Although `spaCy` has released its version 3.0, that comes with language models with improved performance, I used `spaCy` 2.3.5 for the coreference resolution part of this research since `neuralcoref` isn't yet compatible with `spaCy` 3.0.

Before the coreferences could be resolved, the data had to be processed by a dependency parser, a part-of-speech (POS) tagger and a Named Entity Recognition (NER) system. A dependency parser analyzes the grammatical structure of sentences in the data and subsequently, tags each word with its grammatical role, POS tagger tags each word with its POS, and a NER system identifies the available entities in the data. `spaCy` offers several English language models for these tasks.

I used the `en_core_web_lg` model. This is a large model that is trained and eval-

[1] https://github.com/huggingface/neuralcoref

uated on the OntoNotes Release 5.0 dataset [2]. This dataset consists of newswire, broadcast news, broadcast conversation, telephone conversation, and web data. Table 4.1 shows the performance of this language model on the relevant tasks [3], evaluated on the OntoNotes Release 5.0 dataset.

Table 4.1: Performance of spaCy's en_core_web_lg language model [3]

Pipeline	Accuracy	Precision
Dependency parsing	0.92	
Named Entity Recognition		0.86
Part-of-speech tagging	0.97	

Hereafter the data was ready to be processed by `neuralcoref`. This is a pre-trained neural network that aims to detect coreference clusters using word embeddings. The network is also trained on the OntoNotes 5.0 dataset, which is currently the largest coreference annotated dataset in existence. The implementation makes it possible to configure some of the hyper parameters of the network, like the greedyness (the higher this value, the more coreferences are resolved) and a boolean 'blacklist', to exclude a predefined list of words from being resolved. This list contains the following words: 'I', 'me', 'my', 'you', and 'your'. The disadvantage to this blacklist parameter is that the only option is to exclude certain possessive adjectives (the ones in the predefined blacklist) instead of all of them. To solve this, I used an addition to the `neuralcoref` [4], which made it possible to provide the model with a custom blacklist, this way adding the missing possessive pronouns to the blacklist. The advantage of excluding certain words is that possessive adjectives ('my' and 'your') will not be resolved. I will illustrate the resolution of possessive pronouns and the superfluity of it for this research using Example 4.1.

[2] https://catalog.ldc.upenn.edu/LDC2013T19

[3] https://spacy.io/models/

[4] https://github.com/Keating950/neuralcoref

Example 4.1

a. Harry raced up to Gryffindor Tower, collected his Nimbus Two Thousand, and joined the large crowd swarming across the grounds, but his mind was still in the castle along with the bodiless voice, and as he pulled on his scarlet robes in the locker room, his only comfort was that everyone was now outside to watch the game.

b. Harry raced up to Gryffindor Tower, collected Harry Nimbus Two Thousand, and joined the large crowd swarming across the grounds, but Harry mind was still in the castle along with the bodiless voice, and as Harry pulled on Harry scarlet robes in the locker room, Harry only comfort was that everyone was now outside to watch the game.

If the possessive pronouns ('his') in Example 4.1a were resolved, the outcome would be Example 4.1b. However, this will not change the entities in the sentences: 'his mind' will become 'Harry mind', but it will still be a different entity than 'Harry' and it will not have an impact on the resulting entity grid, as I will explain more elaborately in Chapter 5.

It is also possible to provide the model with a conversion dictionary. The model uses this dictionary to learn the embeddings of rare words (like names) by averaging the embeddings of a list of more common words. I used this conversion dictionary to provide the model with a list of common names and words that are specific for the Harry Potter books. One limitation to this conversion dictionary is that it's currently not possible to provide the model with word groups (like a first and a surname); it's only possible to provide the model with single words. The configuration for `neuralcoref`, including this conversion dictionary and the custom blacklist, can be found in Appendix A.

4.2 Results

To evaluate the performance of the application of coreference resolution on the handwritten books, I manually checked the replacements that were performed for two randomly chosen chapters. I subsequently computed the precision by dividing the number of correct replacements by the total number of replacements. For both chapters, the precision of the coreference resolution was 0.69.

I did the same for two chapters of the generated books. As these chapters were shorter on average, less coreferences were resolved. However, the precision of the coreference resolution did not differ greatly from the precision of the coreference resolution on the generated books chapters: for both chapters, the precision was 0.65.

The performance of the coreference resolution algorithm on the books datasets are shown in table 4.2.

Table 4.2: Performance of coreference resolution on handwritten and generated books chapters

Dataset	Word count	Coreferences resolved	Correct resolutions	Precision
Handwritten	6348	214	148	69%
Generated	644	34	22	65%

Next to the performance of the algorithm, it is worth the effort to look into the performed coreference resolution more closely. Example 4.2 comes from the handwritten books dataset. The coreferences that were resolved by the algorithm are highlighted in yellow. Note that not all possible resolutions were found by the algorithm, but all coreferences that were resolved are correctly resolved.

Example 4.2

Original

Hermione hadn't entered the classroom, yet Harry knew she had been right next to him when he had opened the door. "That's weird," said Harry, staring at Ron. "Maybe — maybe she went to the bathroom or something?" But Hermione didn't turn up all lesson. "She could've done with a Cheering Charm on her too," said Ron as the class left for lunch, all grinning broadly — the Cheering Charms had left them with a feeling of great contentment. Hermione wasn't at lunch either. By the time they had finished their apple pie, the after-effects of the Cheering Charms were wearing off, and Harry and Ron had started to get slightly worried. "You don't think Malfoy did something to her?" Ron said anxiously as they hurried upstairs toward Gryffindor Tower. They passed the security trolls, gave the Fat Lady the password ("Flibbertigibbet"), and scrambled through the portrait hole into the common room. Hermione was sitting at a table, fast asleep, her head resting on an open Arithmancy book.

Coreferenced

Hermione hadn't entered the classroom, yet Harry knew Hermione had been right next to Harry when Harry had opened the classroom door. "That's weird," said Harry, staring at Ron. "Maybe — maybe Hermione went to the bathroom or something?" But Hermione didn't turn up all lesson. "Hermione could've done with a Cheering Charm on Hermione too," said Ron as the class left for lunch, all grinning broadly — the Cheering Charms had left them with a feeling of great contentment. Hermione wasn't at lunch either. By the time they had finished their apple pie, the after-effects of the Cheering Charms were wearing off, and Harry and Ron had started to get slightly worried. "You don't think Malfoy did something to Hermione?" Ron said anxiously as they hurried upstairs toward Gryffindor Tower. They passed the security trolls, gave the Fat Lady the password ("Flibbertigibbet"), and scrambled through the portrait hole into the common room. Hermione was sitting at a table, fast asleep, her head resting on an open Arithmancy book.

The algorithm did also make mistakes. Hereafter, I will discuss some of the common mistakes that were made by the system.

Blacklisted words were resolved anyway

In some cases, words that were blacklisted were resolved anyway. An example of such a case can be seen in Example 4.3.

Example 4.3

Original

Gilderoy Lockhart was sitting there, humming placidly to himself. ”His memory’s gone,” said Ron. ”The Memory Charm backfired. Hit him instead of us. Hasn’t got a clue who he is, or where he is, or who we are. I told him to come and wait here. He’s a danger to himself.” Lockhart peered good-naturedly up at them all. ”Hello,” he said. ”Odd sort of place, this, isn’t it?

Coreferenced

Gilderoy Lockhart was sitting there, humming placidly to Gilderoy Lockhart. ”His memory’s gone,” said Ron. ”The Memory Charm backfired. Hit Gilderoy Lockhart instead of us. Hasn’t got a clue who Gilderoy Lockhart is, or where Gilderoy Lockhart is, or who we are. I told Gilderoy Lockhart to come and wait here. Gilderoy Lockhart’s a danger to Gilderoy Lockhart.” Lockhart peered good-naturedly up at them all. ”Hello,” Gilderoy Lockhart said. ”Odd sort of place, this, isn’t it?

Although the resolved coreferences in Example 4.3 are correct, it is striking that the word ‘himself’ was resolved twice even though it is a blacklisted word, meaning that it shouldn’t be resolved.

Excessive inclusion of subordinate clauses

Example 4.4 shows that in some cases, the coreference resolution system replaced the references not only with the correct entity, but it includes an unnecessary subordinate clause with it. The first resolution in Example 4.4 is correct, but for the second and third, only 'Dumbledore' would have sufficed (and would have been correct).

Example 4.4

Original

Merope was left alone in London, expecting the baby who would one day become Lord Voldemort." "How do you know she was in London, sir?" "Because of the evidence of one Caractacus Burke," said Dumbledore, "who, by an odd coincidence, helped found the very shop whence came the necklace we have just been discussing." He swilled the contents of the Pensieve as Harry had seen him swill them before, much as a gold prospector sifts for gold.

Coreferenced

Merope was left alone in London, expecting the baby who would one day become Lord Voldemort." "How do you know Merope was in London, sir?" "Because of the evidence of one Caractacus Burke," said Dumbledore, "who, by an odd coincidence, helped found the very shop whence came the necklace we have just been discussing." Dumbledore, "who, by an odd coincidence, helped found the very shop whence swilled the contents of the Pensieve as Harry had seen Dumbledore, "who, by an odd coincidence, helped found the very shop whence swill them before, much as a gold prospector sifts for gold.

4.3 Limitations and recommendations for future work

The performance of the coreference resolution algorithm was hard to evaluate. I ultimately decided to evaluate the performance by computing the precision of the algorithm. I counted all resolved coreferences and subsequently, took the ratio of the correctly resolved ones. However, hereby, I neglected the coreferences that could have been resolved but weren't.

I took a sample to evaluate the performance of the algorithm, since this was a manual task and it would be too time consuming to evaluate all the data. However, it could be possible that this sample was not representative for the rest of the data. There are evaluation metrics that could assist in the automatic evaluation of the performance of coreference resolution, such as B^3 and $CEAF$ [4]. These metrics

could make it possible to not only evaluate the performance of just a sample of the documents, but all of them.

Coreference resolution is currently a popular topic in NLP. Next to HuggingFace, the organization behind `neuralcoref`, other institutions focused on NLP research are currently working on algorithms to resolve coreferences. Explosion, the organization behind `spaCy`, is actually working together with HuggingFace on developing a coreference resolution module that is part of the core `spaCy` library [5] [6]. I have been monitoring these developments, but unfortunately, new releases were not scheduled in time for this study. However, I expect that in the near future, coreference resolution algorithms will be improved. It could be worth monitoring these developments concerning coreference resolution, and conducting this experiment again when new features and improvements have been released.

4.4 Conclusion

Coreference resolution using `neuralcoref` proved to be moderately successful: evaluating the performance of the algorithm using a sample resulted in a precision score of 65 - 69%. The algorithm did make some repeating mistakes. In the next experiments, I will investigate whether resolving coreferences in this way is of value for the analysis of discourse coherence.

[5] https://github.com/huggingface/neuralcoref/issues/295
[6] https://github.com/explosion/spaCy/pull/7264

Chapter 5

Analyzing syntactic coherence

This chapter describes the experiment that I conducted in order to analyze the syntactic coherence of the documents in all datasets. First, I explain how I applied Barzilay and Lapata's entity grid model [1] to extract syntactic coherence features from the documents in all datasets in Section 5.1, after which I show and discuss the results in Section 5.2 and Section 5.3.

5.1 Methodology

For the automatic assessment of syntactic coherence, I used the entity grid model by Barzilay and Lapata [1] to extract syntactic coherence features. After that, I will use these features to classify the documents into their coherence class.

5.1.1 Extracting syntactic coherence features

As described in Section 2.1.2, an entity grid describes whether an entity occurs in a sentence and if so, it states the grammatical role of the entity in that discourse. Each entity that occurs at least once in a discourse is included in the entity grid. The grammatical roles under consideration for this model are Object ('O'), Subject ('S'), Other ('X'), and not present in sentence ('-').

However, in order to be able to do this meaningfully, the data needed to be parsed with a dependency parser. I parsed the data of the GCDC datasets (original and two point scale) and the short snippets of the books dataset using the `spaCy` dependency parser [1], trained on the English Transformer language model `en_core_web_trf`. This model is preferrable over the language model I used in Section 4.1 (`en_core_web_lg`), since although the latter is optimized for CPU, its accuracy for dependency parsing is slightly lower than that of the `en_core_web_trf` language model (92% versus 95%). Seeing that dependency parsing is one of the most important steps for creating an

[1] https://spacy.io/api/dependencyparser

accurate entity grid, I preferred accuracy over efficiency, and therefore, I used the `en_core_web_trf` pipeline.

The `en_core_web_trf` model uses a RoBERTa base Transformer model. This model is based on Googles BERT model but it is more extensively trained and there were some tweaks to the hyperparameters which resulted in a more robust version of BERT [18]. The model is trained on the OntoNotes Release 5.0 dataset.

Next to this, `spaCy` offers additional pipeline components and functions that can be added to the pipeline in use. One of these is the function `merge_noun_chunks`. This function learns noun chunks (multiple word tokens) from a text that is tokenized to single word tokens. This could be beneficial for the creation of entity grids, and example 5.1 illustrates that.

Example 5.1

Next they were hailed by Ernie Macmillan, a Hufflepuff fourth year, and a little farther on they saw Cho Chang, a very pretty girl who played Seeker on the Ravenclaw team. She waved and smiled at Harry, who slopped quite a lot of water down his front as he waved back. More to stop Ron from smirking than anything, Harry hurriedly pointed out a large group of teenagers whom he had never seen before. “Who d’you reckon they are?” he said. “They don’t go to Hogwarts, do they?” “ ’Spect they go to some foreign school,” said Ron. “I know there are others. Never met anyone who went to one, though. Bill had a pen-friend at a school in Brazil..

Without merging noun chunks, the words ‘Ernie’ and ‘Macmillan’ would be tokenized to two separate words, but they are in fact used to describe the same entity. A benefit of tokenizing this entity into two different tokens is that if in future references only his first name would be used, the repetition of an entity throughout sentences is better captured. However, since both tokens refer to the same entity, it is more correct to include the full entity (all the words) in the entity grid. The repetition problem can be solved by applying coreference resolution (Chapter 4).

After the dependency parsing was completed, I converted each parsed document into an entity grid using TRUNAJOD’s Python implementation [2]. This library instantiates an empty entity map and an empty entity grid (both are empty dictionaries). It subsequently takes the dependency parsed documents as input and loops over each sentence, creating a key per sentence in the empty entity map. Next, it loops over each token, first retrieving its POS tag (which has also been learned using the `spaCy` pipeline). If this POS tag is either a noun (NOUN), pronoun (PRON) or proper noun (PROPN), the token and its dependency tag (as a tuple) are added

[2] https://trunajod20.readthedocs.io/en/latest/api_reference/entity_grid.html

as value to the entity map. If it is the first time that the token occurs in the document, it is added to the entity grid too, with the text of the token as key and a list as long as the number of sentences in the document as value. Each index of the list is initially filled with a dash ('-').

The next step is to fill the entity grid with the correct dependency tags: if a token occurs in a sentence corresponding to the column of the grid, the '-' is replaced with the correct grammatical role's dependency tag. In order to do this, the dependency tag of the token is converted to the correct entity grid tag, either 'X', 'O' or 'S', using the conversion rules from Table 5.1 [2]. It should be noted that if an entity occurs more than once in a sentence, the entity tag that is highest in rank will be used in the grid. The hierarchical order of entity tags is S, O, X, -, from high to low.

Table 5.1: Conversion rules - Entity Grid tags and their corresponding dependency tags [2]

Entity Grid tag	Dependency tag
S	nsub, csubj, csubjpass, dsubjpass
O	iobj, obj, pobj, dobj
X	For any other dependency tag

This resulted ultimately in an entity grid per document. From this entity grid, I was able to compute the corresponding feature vector Φ, which consists of 16 dimensions. Each dimension represents the probability p_n of a specific sentence-to-sentence transition of entities in the corresponding entity grid. For example: if an entity is object in the first sentence, and subject in the next, the sentence-to-sentence transition is 'OS' (Object Subject). The probability that a specific transition occurs is simply computed by its ratio: the number of times that specific transition occurs, divided by the total number of transitions in the document. Following this, a feature vector is composed as follows:

$$\Phi = (p_1(--), p_2(-O), p_3(-S), p_4(-X), p_5(O-), p_6(OO), p_7(OS), p_8(OX),$$
$$p_9(S-), p_{10}(SO), p_{11}(SS), p_{12}(SX), p_{13}(X-), p_{14}(XO), p_{15}(XS), p_{16}(XX))$$

5.1.2 Classification

Ultimately, for each document in each dataset, I ended up with a feature vector containing the 16 sentence-to-sentence probabilities, resulting from the entity grid of the document. These values served as input to two classification models. I used two models from Python's `scikit-learn` Machine Learning library [3], namely logistic

[3] https://scikit-learn.org/

regression (LR) and random forest (RF). The models were selected for their good trade-off between accuracy and interpretability. The latter made it possible to find the most important features for the coherence decision afterwards by computing feature importances (RF) or finding the coefficients of the intercept (LR) [19] [26]. One remark should be made regarding the interpretability of RF, since this model is not directly interpretable - as is the case with LR. However, it is possible to compute feature importances for an RF by computing the mean decrease impurity over all trees in the RF (as I will explain in Section 5.2.3), and therefore, I chose to include this model as well.

The RF model was initiated with the default value of 100 decision trees (DTs), and each DT had a maximum tree depth of 5: a relatively low value, to prevent the model from overfitting on the training data. The DTs used the Gini criterion to determine the best split because it is computationally efficient, as it does not require logarithmic computations. The LR model was initiated with the default values of `sklearn`'s LogisticRegressionClassifier, except for the maximum number of iterations that the model could use to converge. The default of this value is 100, but I used a value of 1000 iterations to enlarge the probability that the model would converge.

5.2 Results

I will show the results of this experiment using an example. The syntactic coherence in the documents was analyzed by means of retrieving the feature vectors containing the sentence-to-sentence transition probabilities of the syntactic roles of the entities in the documents. In order to do that, the first step was to create an entity grid per document, for all documents in the GCDC original data, the GCDC two-point scale data, and the short snippets of the books data. An entity grid is of size N x M, in which N is the number of entities in a document, and M is the number of sentences the document contains. The header row is filled with the available entities in the documents; the header column is filled with numbers that represent the sentences in the document. Note that this entity grid is thus transposed compared to the entity grid from the original paper (Table 2.1), in which the rows represented the sentences and the columns represented the available entities. The cells of the entity grids are filled with the syntactic role of the corresponding entity in the corresponding sentence if the entity is present, else its cell is filled with a dash ('-'). The entity grids were created for both the original documents and the documents in which coreferences were resolved. Figure 5.1 shows an example of an entity grid of the original text. Figure 5.2 shows the entity grid of the same document, but after applying coreference resolution.

1. Dobby stopped in front of the brick fireplace and pointed.
2. ”Winky, sir!” he said.
3. Winky was sitting on a stool by the fire.
4. Unlike Dobby, she had obviously not foraged for clothes.
5. She was wearing a neat little skirt and blouse with a matching blue hat, which had holes in it for her large ears.
6. However, while every one of Dobby’s strange collection of garments was so clean and well cared for that it looked brand-new, Winky was plainly not taking care of her clothes at all.
7. There were soup stains all down her blouse and a burn in her skirt.
8. ”Hello, Winky,” said Harry.
9. Winky’s lip quivered.

	1	2	3	4	5	6	7	8	9
Dobby	S	-	-	O	-	-	-	-	-
front	O	-	-	-	-	-	-	-	-
the brick fireplace	O	-	-	-	-	-	-	-	-
Winky	-	X	S	-	-	S	-	X	-
sir	-	X	-	-	-	-	-	-	-
he	-	S	-	-	-	-	-	-	-
a stool	-	-	O	-	-	-	-	-	-
the fire	-	-	O	-	-	-	-	-	-
she	-	-	-	S	S	-	-	-	-
clothes	-	-	-	O	-	-	-	-	-
a neat little skirt	-	-	-	-	O	-	-	-	-
blouse	-	-	-	-	X	-	-	-	-
a matching blue hat	-	-	-	-	O	-	-	-	-
holes	-	-	-	-	O	-	-	-	-
it	-	-	-	-	O	S	-	-	-
her large ears	-	-	-	-	O	-	-	-	-
Dobby’s strange collection	-	-	-	-	-	O	-	-	-
garments	-	-	-	-	-	O	-	-	-
brand	-	-	-	-	-	X	-	-	-
care	-	-	-	-	-	O	-	-	-
her clothes	-	-	-	-	-	O	-	-	-
there	-	-	-	-	-	-	X	-	-
soup stains	-	-	-	-	-	-	X	-	-
her blouse	-	-	-	-	-	-	O	-	-
a burn	-	-	-	-	-	-	X	-	-
her skirt	-	-	-	-	-	-	O	-	-
Harry	-	-	-	-	-	-	-	S	-
Winky’s lip	-	-	-	-	-	-	-	-	S

Figure 5.1: An original document with its corresponding entity grid

1. Dobby stopped in front of the brick fireplace and pointed.
2. ”Winky, sir!” Dobby said.
3. Winky was sitting on a stool by the fire.
4. Unlike Dobby, Winky had obviously not foraged for clothes.
5. Winky was wearing a neat little skirt and blouse with a matching blue hat, which had holes in it for her large ears.
6. However, while every one of Dobby’s strange collection of garments was so clean and well cared for that it looked brand-new, Winky was plainly not taking care of her clothes at all.
7. There were soup stains all down her blouse and a burn in her skirt.
8. ”Hello, Winky,” said Harry.
9. Winky’s lip quivered.

	1	2	3	4	5	6	7	8	9
Dobby	S	S	-	O	-	-	-	-	-
front	O	-	-	-	-	-	-	-	-
the brick fireplace	O	-	-	-	-	-	-	-	-
Winky	-	S	S	S	S	S	-	X	-
sir	-	X	-	-	-	-	-	-	-
a stool	-	-	O	-	-	-	-	-	-
the fire	-	-	O	-	-	-	-	-	-
clothes	-	-	-	O	-	-	-	-	-
a neat little skirt	-	-	-	-	O	-	-	-	-
blouse	-	-	-	-	X	-	-	-	-
a matching blue hat	-	-	-	-	O	-	-	-	-
holes	-	-	-	-	O	-	-	-	-
it	-	-	-	-	O	S	-	-	-
her large ears	-	-	-	-	O	-	-	-	-
Dobby’s strange collection	-	-	-	-	-	O	-	-	-
garments	-	-	-	-	-	O	-	-	-
brand	-	-	-	-	-	X	-	-	-
care	-	-	-	-	-	O	-	-	-
her clothes	-	-	-	-	-	O	-	-	-
there	-	-	-	-	-	-	X	-	-
soup stains	-	-	-	-	-	-	X	-	-
her blouse	-	-	-	-	-	-	O	-	-
a burn	-	-	-	-	-	-	X	-	-
her skirt	-	-	-	-	-	-	O	-	-
Harry	-	-	-	-	-	-	-	S	-
Winky's lip	-	-	-	-	-	-	-	-	S

Figure 5.2: The document from Figure 5.1 after applying coreference resolution and its corresponding entity grid

From these entity grids, the feature vectors per document could be computed. These feature vectors consist of the 16 sentence-to-sentence transition probabilities: each vector dimension represents the probability of a transition type. A feature vector is composed as follows:

$$\Phi = (p_1(--), p_2(-O), p_3(-S), p_4(-X), p_5(O-), p_6(OO), p_7(OS), p_8(OX),$$
$$p_9(S-), p_{10}(SO), p_{11}(SS), p_{12}(SX), p_{13}(X-), p_{14}(XO), p_{15}(XS), p_{16}(XX))$$

This means, the first dimension of the feature vector, $p_1(--)$, represents the probability p that an entity is absent in one sentence ('-'), and it is also absent in the next sentence ('-'). This sentence-to-sentence transition probability is computed by dividing the number of times that this transition is present in the document, by the total number of transitions in the document.

Following this, the feature vector of the example in Figure 5.1 resulted in this vector:

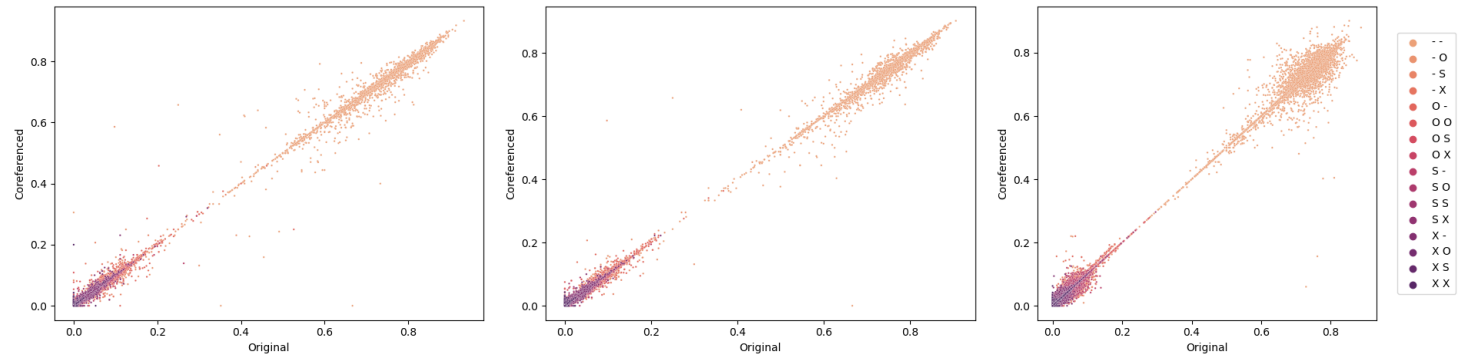
$$\Phi = (0.728, 0.067, 0.022, 0.036, 0.071, 0.000, 0.004, 0.000,$$
$$0.031, 0.000, 0.004, 0.000, 0.031, 0.000, 0.004, 0.000)$$

The feature vector corresponding to the example in Figure 5.2 resulted as follows:

$$\Phi = 0.721, 0.072, 0.014, 0.034, 0.077, 0.000, 0.005, 0.000,$$
$$0.019, 0.000, 0.024, 0.000, 0.034, 0.000, 0.000, 0.000$$

5.2.1 Coreference resolution and syntactic coherence features

To investigate the influence of coreference resolution on the syntactic coherence features, I plotted the value of the syntactic coherence features of the original doc-



(a) GCDC Original *(b) GCDC two-point scale* *(c) Short snippets of books*

Figure 5.3: Correlation between the syntactic coherence features of the original documents and the syntactic coherence features of the documents in which coreferences were resolved

uments against the values of the syntactic coherence features of the same documents in which coreferences were resolved. I did this for each dataset separately. The resulting scatter plots are shown below in Figure 5.3.

Figure 5.3 shows that the syntactic coherence features (the 16 sentence-to-sentence transition probabilities) of the original documents of each dataset correlate linearly with the syntactic coherence features of the documents in which coreferences were resolved. This indicates that the values of the syntactic coherence features (sentence-to-sentence transition probabilities) do not change greatly if coreference resolution is applied on the data.

5.2.2 Classification using syntactic coherence features

The goal was to provide two models, the RF model and the LR model, with the syntactic coherence features of the documents, and to have them correctly predict the coherence label that corresponds to the document based on these features. The performance of the models was evaluated by computing their accuracy score. For the RF model, I also computed the Out of Bag (OOB) score. I will first briefly explain both these metrics before I proceed to the results of the classification.

The *accuracy* is the fraction of all correctly predicted datapoints among all predictions that were made. It can be evaluated by dividing the number of true positives TP and true negatives TN by the total number of predictions (true positives TP, true negatives TN, false positives FP and false negatives FN):

$$\text{Accuracy } = \frac{TP + TN}{TP + TN + FP + FN}$$

This metric can be misleading if the dataset is imbalanced, but since I handled the class imbalance as described in Chapter 3, it is a good metric to evaluate the performance of the model.

To explain the *OOB score*, it is important to understand the workings of an RF. An RF is composed of a number of DTs. In this case, the RF consists of 100 trees with a depth of 5. Each tree is separately trained by providing it with a randomly generated subset of the training data (called a 'bootstrap sample'). Ultimately, the prediction of the RF on a datapoint is basically the consensus between the DTs on that datapoint: the majority rules. The set of datapoints that wasn't part of the bootstrap sample is called the OOB sample. Since each DT is provided with a unique bootstrap sample as train data, each DT has other datapoints that are OOB. The OOB sample can therefore be used to validate the RF: after the training phase of the DTs has been completed, the OOB samples are provided to the DTs. The prediction of the RF on a datapoint is again based on the majority vote of the individual DTs: the predictions of the DTs for which the datapoint belonged to the OOB sample are counted, and

the RF predicts the class that got the most votes from the DTs. After this has been completed for all OOB datapoints, the OOB score can be computed: it is the fraction of correctly predicted OOB datapoints over all OOB datapoints.

The performance of the models on the syntactic coherence features of the documents of all datasets can be found in Table 5.2.

Table 5.2: Performance of models on syntactic coherence features, on all datasets

Dataset		**Random forest**		**Logistic regression**
		Accuracy	OOB	Accuracy
GCDC original	*Original*	0.39	0.36	0.37 *
	Coreferenced	0.41	0.38	0.37 *
GCDC two-point scale	*Original*	0.82	0.83	0.69
	Coreferenced	0.83	0.83	0.71
Short snippets of books	*Original*	0.81	0.80	0.72
	Coreferenced	0.79	0.79	0.72

** Model did not converge*

Table 5.2 shows that the RF model made the most accurate predictions on the GCDC two-point scale dataset and on the short snippets of the books. The model performed worse on the GCDC original dataset. However, it is important to remember that this dataset did not consist of two, but of three classes. If the dataset would consist only of two classes, it would mean that the model performed worse than if it would guess the classes randomly (the probability to predict the class correctly would be 0.5). Since the dataset consists of three classes, the model performs slightly better than it would do randomly (the probability to randomly predict the class correctly with three classes is 0.33). The model shows little to no difference in performance between the original documents of the datasets and the documents in which coreferences were resolved.

The accuracy of the LR model can also be found in Table 5.2. It should be noted that the model did not converge after 1000 iterations on the original GCDC data (neither on the original documents nor on the coreferenced documents).

The table shows that just like the RF model, the LR model performed worst on the three-point-scale GCDC original dataset. Again, the model showed only a small difference to no difference in performance between the original documents of the datasets and the documents in which coreferences were resolved.

5.2.3 Feature importances

Next to the performance of the models, I was interested in the features that the models deemed most important in their decision for predicting a document as either

low or highly coherent. Therefore, I retrieved feature importances for both models, which give an indication of the contribution of a feature to the decision of the model.

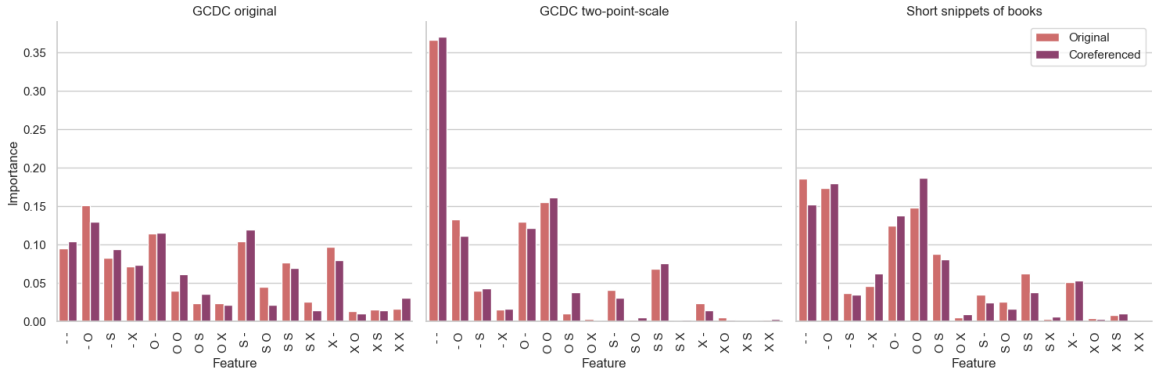
For the RF model, the feature importances can be retrieved by computing the mean decrease impurity of the RF for each feature. The mean decrease impurity is computed by first summing the decrease in node impurity that a feature causes, divided by the number of trees, and weighted by the number of samples a feature splits. This results in a feature importance value in the range from 0 to 1, and all feature importances sum to 1. The higher the value, the more important the feature is. The feature importances of the RF per dataset can be found in Figure 5.4a.

With LR, a feature's importance value is equal to its coefficient value of the intercept. These numbers can be both positive and negative: a positive value means that the feature contributes to the probability for the document to be highly coherent, whereas a negative value means that the feature does the opposite: it contributes to the probability for the document to be low coherent. Other than the values of the feature importances of the RF, the values of the feature importances of the LR do not add up to 1 and they can have every real value. The feature importances for the LR model for all datasets can be found in Figure 5.4b. Note that the feature importances on the GCDC original dataset are very small: that can be attributed to the fact that the model did not converge on this dataset.

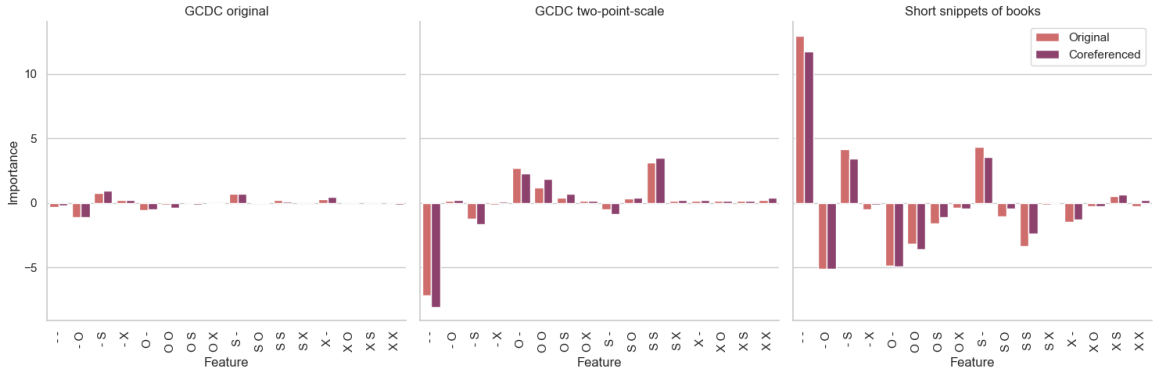
5.3 Discussion

The first thing that strikes when looking at the performance of the models on the different datasets, is that the models perform significantly worse on the GCDC original data then they do on the GCDC two-point scale data and the short snippets of the books. The models perform only a little better than a random classifier would do on the GCDC original data (3-8% better). The reason for this difference can probably be attributed to the difference in the nature of the data. All documents in the GCDC original dataset are written by humans, whereas the low coherent GCDC two-point scale documents were created by me by shuffling sentences randomly, and the low coherent short snippets of books were generated by a computer. This indicates that the models perform better when distinguishing generated texts from handwritten texts using their syntactic coherence features, rather than distinguishing low coherent handwritten texts from highly coherent handwritten texts using their syntactic coherence features.

Furthermore, the classification models show no big difference in performance when using the syntactic coherence features of the original documents, compared to when using the syntactic coherence features of the documents in which coreferences were resolved. This is reflected both in the performance of the models on the



(a) Random forest classifier



(b) Logistic regression model

Figure 5.4: Feature importances of the (a) random forest classifier and the (b) logistic regression model on all datasets

datasets (Table 5.2), and the feature importances of the models (Figure 5.4). This is in contrast with my expectations. I expected that resolving coreferences would cause entities to be repeated throughout sentences, resulting in less entities per document (thus, less rows in the entity grid) but denser rows in the entity grid for prominent entities in highly coherent documents. Subsequently, compared to the original documents, I would have expected that 'O S', 'O O', 'S O', 'S S' transitions would be more important positively, whereas '- -', '- X', 'X -', 'X X' transitions would be more important negatively for the model's coherence decisions. This is not reflected in the feature importances: they show no big difference between the original documents and the documents in which coreferences were resolved. This could be caused by the fact that, as I showed in Chapter 4, not all available coreferences were resolved, and this performance evaluation was carried out on a chapter. In the short snippets, the amount of resolved coreferences was even smaller. It could be possible that too little coreferences were resolved to be of any significant influence.

Comparing the performance of the RF with the LR shows that both models do not perform well on the GCDC original data. On the other two datasets, the accuracy

of the RF is 7-13% higher than the accuracy of the LR. This could be attributed to the fact that an RF can handle more complex decisions, allowing for a split at each node, whereas the LR aims to find an intercept between the classes. However, this makes the RF more likely to overfit on the data than the LR.

When comparing the feature importances of the LR on the GCDC two-point scale data with the feature importances of the LR on the short snippets of the books, it strikes that although the size of most values is comparable, the direction is almost always the opposite. This means that for example, on the GCDC two-point scale data, the '- -' transition contributes strongly in the decision for a document to be low coherent, whereas the same feature contributes strongly for a document to be highly coherent in the short snippets of the books dataset. This could indicate that the coherent classes are labeled oppositely in both datasets, but since that is not the case, the cause must be different. However, in this case, the cause is not clear. A possible explanation could be that the LR converged in a different local minimum, resulting in other values for the intercept and thus, other feature importances.

5.4 Limitations and recommendations for future work

In order to analyze syntactic coherence, I created entity grids per document and I subsequently computed the sentence-to-sentence transition probabilities. I only did this for two sentence transitions: I compared the grammatical role of an entity in the first sentence with the grammatical role of that same entity in the second sentence. A possible extension to this methodology could be to increase this number: compare the grammatical role of an entity in the first sentence with the grammatical role of that same entity in the third, fourth, or fifth sentence. It could be interesting to compare the results and see whether there is an optimal number.

Another addition to the analysis of syntactic coherence using the entity grid model could be to apply weights to the different sentence-to-sentence transitions. I now treated all the transitions equally important, but it could be interesting to reward discourse in which (for example) a lot of 'O S' transitions occur, and see if, and how, that changes the results. In their paper, Barzilay and Lapata propose weights based on the hierarchical order of the syntactic roles, which is, from high to low: S, O, X, - [1].

5.5 Conclusion

Analyzing syntactic coherence using the entity grid model shows interesting results. Based on the sentence-to-sentence transition probabilities that can be derived from a document's entity grid, the RF and LR model could distinguish between gener-

ated low coherent and handwritten highly coherent discourse. However, both models were not able to distinguish between handwritten low coherent discourse and handwritten highly coherent discourse. This indicates that the syntactic coherence features that were obtained using the entity grid model actually help to recognize generated discourse from handwritten discourse, instead of helping to recognize highly coherent discourse from low coherent discourse.

Chapter 6

Analyzing semantic coherence

This chapter describes the experiment that I conducted in order to analyze the semantic coherence of the documents in all datasets. The chapter is structured in the same way as the previous chapter: in Section 6.1, I describe the methodology I used to extract the semantic features from the documents, and subsequently, how I used these features in order to classify the documents in terms of coherence. Thereafter, in Section 6.2, I show the results of this experiment. Finally, I will discuss these results in Section 6.3, explain some limitations to the research in Section 6.4 and I draw some partial conclusions in Section 6.5.

6.1 Methodology

To assess semantic coherence of the datasets, I used word embeddings, based on Section 2.2.2. This means that I retrieved the numerical vector representation for discourse segments, and subsequently, computed the cosine similarity between adjacent discourse segments. However, first, the data had to be preprocessed.

6.1.1 Preprocessing

Prior to creating word embeddings from the datasets, the data was preprocessed by applying lemmatization and removing English stopwords. Stopwords do not add any unique or descriptive information to the meaning of the text, and therefore, will not contribute to the similarity score. For the removal of stopwords, I used `spaCy`'s built-in stopword list.

Lemmatization means to reduce all words to their dictionary form. I did this because the meaning of a word does not change significantly if the word appears in a different form, and therefore, the inflection will not contribute to the semantic similarity. Lemmatization is preferred over stemming, since lemmatization attempts to retrieve the correct POS of a word, its lemma, whereas stemming only reduces a

word to its stem. For lemmatization, I used the lemmatizer that comes with `spaCy`'s language model `en_core_web_lg`.

Example 6.1 shows the difference between a text without any preprocessing (1), with lemmatization (2) and with lemmatization and the removal of stopwords (3).

Example 6.1

1. *Original*
   Harry felt as though he were carrying some kind of talisman inside his chest over the following two weeks, a glowing secret that supported him through Umbridge's classes and even made it possible for him to smile blandly as he looked into her horrible bulging eyes.

2. *Lemmatized*
   Harry feel as though he be carry some kind of talisman inside his chest over the follow two week, a glow secret that support he through Umbridge 's class and even make it possible for he to smile blandly as he look into her horrible bulge eye.

3. *Lemmatized & stopwords removed*
   Harry feel carry kind talisman inside chest follow week, glow secret support Umbridge class possible smile blandly look horrible bulge eye.

6.1.2 Computing semantic similarity

After the preprocessing was completed, I computed the semantic similarity between adjacent discourse segments using word embeddings and the cosine similarity score. For this, I again used the Python library `spaCy`, release 3.0.6 and the pretrained language model `en_core_web_lg`.

This language model includes 684830 unique words and vectors, learned from the OntoNotes 5 dataset, using the Global Vectors (GloVe) model. The vectors, containing 300 dimensions each, are pretrained and static. `spaCy` also offers the `en_core_web_trf` Transformer based language model, which does not include static vectors but learns the context vectors from the dataset under consideration. However, because of time constraints, I preferred the static vectors from the pretrained model. The downside to this is that the model will return an empty vector for each word that is out of vocabulary (OOV), or in other words: it will return an empty vector for each word that is not one of the 684830 keys that is included in the model.
Since I'm not interested in the coherence between the different documents of the GCDC data (original and two point scale), I computed the semantic similarity within each document. I also did this for the short snippets of the books data, to be able to

compare the results on the different datasets afterwards. The first step was to compute a vector for each sentence from the vectors of the words in the sentence. `spaCy` does this by taking the average of the vectors of the relevant word vectors. However, as explained earlier, the model returns an empty vector for each OOV word. Since empty vectors will have a disproportionate effect on the average vector for the sentence, I did not include words that returned an empty vector when computing the vectors for the sentences.

Hereafter, I computed the cosine similarity between the vectors of adjacent sentences and ultimately, as described in [8], I computed the mean cosine similarity (MCS) and the corresponding standard deviation (Std) of these cosines to end up with two semantic coherence features per document: the MCS and corresponding Std.

6.1.3 Classification

The MCS and corresponding Std then served as input to the same classification models as I used for the classification based on syntactic coherence features: the RF and the LR. To be able to compare the classification of the documents based on syntactic coherence features with the classification based on semantic coherence features, the models were initiated with the same parameters as described in Section 5.1.2.

6.2 Results

Like the experiment regarding syntactic coherence, I will describe the results of this experiment concerning the extraction of semantic coherence features using an example. In order to analyze the semantic coherence of the documents in all datasets (GCDC original, GCDC two-point scale, short snippets of books) the semantic similarity between adjacent sentences was computed for both the original data and the data on which coreference resolution was applied. Before this, however, the data had to be preprocessed. For preprocessing, English stopwords were removed and the words in the documents were lemmatized. Words for which the language model did not hold a vector representation were also removed. In Example 6.2, the same documents from Figure 5.1 and Figure 5.2 are shown, before and after preprocessing. If you look closely at both documents after preprocessing, it is clear that the original document contains one less sentence than the one on which coreference resolution was applied: in the latter, the sentence *"Winky sir!", Dobby said.* was regarded as two separate sentences.

Example 6.2

Original document before preprocessing
Dobby stopped in front of the brick fireplace and pointed. ”Winky, sir!”, he said. Winky was sitting on a stool by the fire. Unlike Dobby, Winky had obviously not foraged for clothes. She was wearing a neat little skirt and blouse with a matching blue hat, which had holes in it for her large ears. However, while every one of Dobby’s strange collection of garments was so clean and well cared for that it looked brand-new, Winky was plainly not taking care of her clothes at all. There were soup stains all down her blouse and a burn in her skirt. ”Hello, Winky,” said Harry. Winky’s lip quivered.

Original document after preprocessing
Dobby stop brick fireplace point. Winky sir say. Winky sit stool fire. Unlike Dobby obviously forage clothe. Wear neat little skirt blouse match blue hat hole large ear. Dobby strange collection garment clean care look brand new Winky plainly take care clothe. Soup stain blouse burn skirt. Hello Winky say Harry. Winky lip quiver.

Document with coreferences resolved before preprocessing
Dobby stopped in front of the brick fireplace and pointed. ”Winky, sir!”, Dobby said. Winky was sitting on a stool by the fire. Unlike Dobby, Winky had obviously not foraged for clothes. Winky was wearing a neat little skirt and blouse with a matching blue hat, which had holes in it for her large ears. However, while every one of Dobby’s strange collection of garments was so clean and well cared for that it looked brand-new, Winky was plainly not taking care of her clothes at all. There were soup stains all down her blouse and a burn in her skirt. ”Hello, Winky,” said Harry. Winky’s lip quivered.

Document with coreferences resolved after preprocessing
Dobby stop brick fireplace point. Winky sir. Dobby say. Winky sit stool fire. Unlike Dobby Winky obviously forage clothe. Winky wear neat little skirt blouse match blue hat hole large ear. Dobby strange collection garment clean care look brand new Winky plainly take care clothe. Soup stain blouse burn skirt. Hello Winky say Harry. Winky lip quiver.

Table 6.1: Mean and standard deviation of cosine similarities of adjacent sentences from Example 6.2

	Original	**Coreferenced**
Mean	0.483	0.447
Standard deviation	0.146	0.170

After the preprocessing was completed, the cosine similarity between adjacent sentences was computed. For the original document from Example 6.2, the cosine similarities were as follows:

$$0.303, 0.578, 0.375, 0.499, 0.710, 0.572, 0.299, 0.530$$

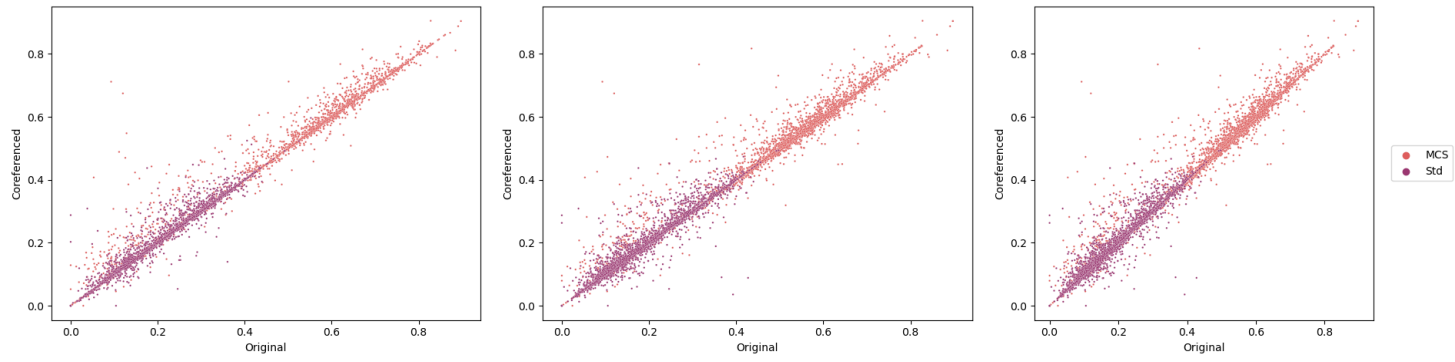
For the document from Example 6.2 after applying coreference resolution, the cosine similarities were:

$$0.189, 0.341, 0.308, 0.531, 0.530, 0.722, 0.572, 0.299, 0.530$$

The final similarity score per document, the metric for semantic coherence, was ultimately computed by taking the MCS and Std of these cosine similarities. The values of these features corresponding to the document in Example 6.2 can be seen in Table 6.1.

6.2.1 Coreference resolution and semantic coherence features

To investigate the influence of the coreference resolution on the semantic coherence features, I plotted the value of the semantic coherence features of the original



(a) GCDC Original *(b) GCDC two-point scale* *(c) Short snippets of books*

Figure 6.1: Correlation between the semantic coherence features of the original documents and the semantic coherence features of the documents in which coreferences were resolved

documents against the values of the semantic coherence features of the documents in which coreferences were resolved. I did this for each dataset separately. The resulting scatter plots can be found in Figure 6.1.

The figure shows that the semantic coherence features of the original documents and the documents on which coreference resolution was applied correlate linearly, as was the case with the syntactic coherence features. This indicates that also for the semantic coherence features, coreference resolution does not cause a big change in values.

6.2.2 Classification using semantic coherence features

After the semantic coherence features were extracted, I provided both classification models (random forest (RF) and logistic regression (LR)) with these features, and I again evaluated the models by computing the accuracy and OOB score. The performance of the models on the semantic coherence features of the documents of all datasets can be found in Table 6.2.

Table 6.2: Performance of models on semantic coherence features, on all datasets

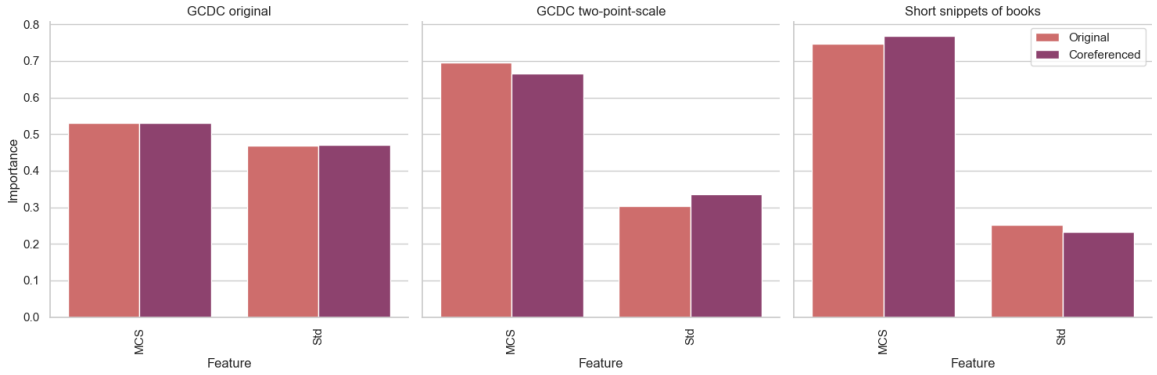
Dataset		**Random forest**		**Logistic regression**
		Accuracy	OOB	Accuracy
GCDC original	*Original*	0.36	0.36	0.35
	Coreferenced	0.34	0.36	0.36
GCDC two-point scale	*Original*	0.83	0.84	0.65
	Coreferenced	0.84	0.84	0.67
Short snippets of books	*Original*	0.66	0.65	0.54
	Coreferenced	0.68	0.66	0.55

Table 6.2 shows that again, both models do not perform well on the GCDC original data: they perform only 1-3% better than a random classifier would perform on this data. This is also the case for the LR on the short snippets of the books data: the model only performs 4-5% better than would be expected from a random classifier on a two class dataset. The models both perform better on the GCDC two-point scale data.

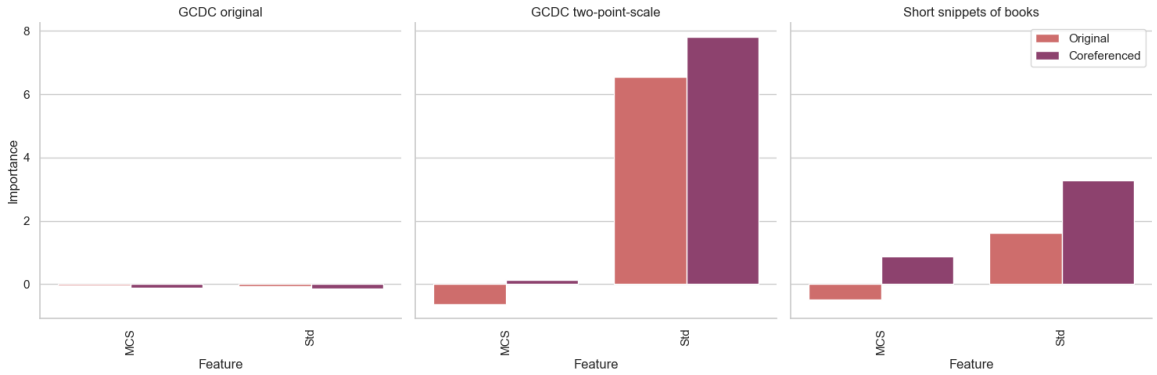
The results in Table 6.2 also show no significant difference in performance between the original data and the data in which coreferences were resolved.

6.2.3 Feature importances

Figure 6.2 shows the feature importances of both models on the semantic coherence features of all datasets. Here, a small but remarkable difference between the original documents and the documents in which coreferences were resolved can be found in



(a) Random forest classifier



(b) Logistic regression model

Figure 6.2: Feature importances of the (a) random forest classifier and the (b) logistic regression model on the semantic coherence features of all datasets

the feature importances of the LR: on the GCDC two-point scale data and the short snippets of the books data, the MCS has a negative influence on the coherence decision for the original documents, but it has a positive influence on the coherence decision for the documents in which coreferences were resolved.

6.3 Discussion

Like the results of the syntactic coherence analysis, the results of this semantic coherence analysis show that the models both perform worst on the GCDC original data. As I explained in Section 5.3, this can probably be attributed to the fact that the GCDC original data consists only of data that has been written by humans, whereas the other datasets contain data that has been manipulated or generated by a computer. This thus indicates that the models can distinguish handwritten documents from generated documents based on their semantic coherence features, but perform worse when distinguishing low coherent handwritten documents from highly

coherent handwritten documents based on their semantic coherence features.

The LR model did not perform well on the semantic coherence features of the short snippets of the books data: the model's accuracy was only 5% higher than you would expect from a classifier that has to classify data into two classes randomly. With an accuracy of 66%, the RF performed better, but it is still not as good as the performance of the models on the syntactic coherence features. It is striking that the models performed worse on the short snippets of the books data than they did on the GCDC two-point scale data, as these datasets both contain generated or manipulated documents versus handwritten documents. This was not the case with the classification based on the syntactic features of the dataset. A probable explanation could be that the low coherent data in the GCDC two-point scale dataset is manipulated data (the sentences of all documents were shuffled), whereas the low coherent data in the short snippets of the books dataset is generated by a system. It could be that this different way of creating the data is the cause for the difference in performance between these two datasets. These results, compared to the performance of the classification based on the syntactic coherence features (Chapter 5), could indicate that the low coherent documents from the short snippets of the books (the generated books from NaNoGenMo), are more coherent semantically they are syntactically. That is, the models can clearly distinguish between the low coherent documents and the highly coherent documents in this dataset when using the syntactic coherence features, but they can hardly do so when provided with the semantic coherence features of the dataset.

The difference in performance between classification based on semantic coherence features and classification based on syntactic coherence features, could be caused by the way the semantic coherence features were computed: by taking the cosine similarity of adjacent sentences within a document, and then, computing the mean of these cosine similarities (MCS). However, as the cosine similarities from Example 6.2 show, the cosine similarities within a document could differ greatly. The length of sentences could be very different, especially after preprocessing the data. It could happen that the first sentence did not contain any stopwords, whereas the second sentence consisted mainly of stopwords. The first sentence would be considerably longer than the second sentence, which would mean that because of the fact that the vector of a sentence is an average of the vectors of the words it contains, the first sentence loses more information than the second sentence. The length of the sentences and the number of sentences in a document could have a disproportionate influence on the MCS, which could cause for a validation bias. I tried to account for this variation in cosine similarities by computing the Std next to the MCS, but this appeared not to be sufficient. I expect that the bias will decrease when computing the cosine similarity between longer discourse segments, because

the difference in length between discourse segments will become relatively smaller.

When looking at the feature importances of the models on the semantic coherence features of the documents, it strikes that the MCS of the original documents of the GCDC two-point scale data and the original documents of the short snippets of the books data has a negative influence on the coherence decision of the models, whereas the MCS of the documents in which coreferences are resolved has a positive influence on the coherence decision. This means that for the original documents, a higher MCS indicates a low coherent document, whereas for the documents in which coreferences were resolved, a high MCS indicates a highly coherent document. The latter can probably be attributed to the fact that documents in which coreferences were resolved contain more duplicate words than the original documents, which will increase the similarity of the documents. This will be the case for both the low coherent documents and the highly coherent documents, but as I showed in Chapter 4, more coreferences were resolved in the highly coherent documents than in the low coherent documents, which could cause this difference in feature importance.

6.4 Limitations and recommendations for future work

In this semantic coherence experiment, I used a lot of averages. First: to compute the similarity between adjacent sentences, the sentences had to be embedded in the semantic space. To do that, I had to establish the vectors that approximate the meaning of the sentences. I did this by taking the average of the words in the sentence (leaving out stopwords and words without a vector). Hereafter, I computed the cosine similarities of adjacent sentences. To ultimately end up with one cosine value per document, I took the average of the resulting cosines (MCS). However, this means that I did lose some information. This was illustrated by the Std corresponding to the MCS, which I also computed: this Std was often relatively high, which means that the separate cosines per document were spread out compared to the MCS, and that the similarity between adjacent sentences often differed greatly. This could be caused by the length of sentences, which could be very different, especially after preprocessing the data. It could happen that the first sentence did not contain any stopwords, whereas the second sentence consisted mainly of stopwords. The first sentence would be considerably longer than the second sentence, which would mean that because of the fact that the vector of a sentence is an average of the vectors of the words it contains, the first sentence loses more information than the second sentence. I did not account for the length of the sentences when looking at the similarity between them.

I did also not account for the number of sentences within a document when

computing the final cosine similarity score of a document. As shown in Example 6.2, the number of sentences in a document could be different between documents, even between the original version of a document and the same document in which coreferences are resolved. The language model that I used does not seem to be consistent in determining the end of a sentence.

The difference in performance when using the syntactic features of the short snippets of the books data, compared to when using the semantic coherence features of the same data, raises the question: is it possible that Natural Language Generation (NLG) systems generate discourse that is more coherent syntactically than it is semantically? Considering the amount of data used in this research, it is a bit early to draw conclusions. It could be an interesting topic for future research, in which more data is used, and from more different sources than only the seven books in this research.

As explained in Section 6.1.2, I used static vectors from the `en_core_web_lg` language model. However, because of this, some words were not represented by a vector that was available in this language model. A solution to this would be to use a Transformer based language model, like `en_core_web_trf`. This models learns word vectors from the context of the words in the dataset.

6.5 Conclusion

Analyzing semantic coherence using the MCS and corresponding Std of adjacent sentences shows mixed results. The RF model and LR model clearly distinguish highly coherent data from low coherent data in the GCDC two-point scale dataset, but they perform worse on the short snippets of the books data and bad on the GCDC original data. Classification based on just the semantic coherence features seems less promising than classification based on just the syntactic coherence features of the documents.

Chapter 7

Syntactic and semantic coherence: a synthesis

The next step was to combine the syntactic and semantic coherence features of the documents of all datasets and provide the classification models with all of these features. The syntactic coherence features were already extracted as described in Chapter 5, and the semantic coherence features were extracted as described in Chapter 6. In this chapter, I will therefore not describe the methodology again, but immediately proceed to the results of the classification of the documents in all datasets using both the syntactic and semantic coherence features.

7.1 Results

Table 7.1 shows the results of the classification using the random forest (RF) and the logistic regression (LR) on the syntactic and semantic coherence features of the documents in all datasets.

Table 7.1: Performance of models on syntactic and semantic coherence features of documents in all datasets

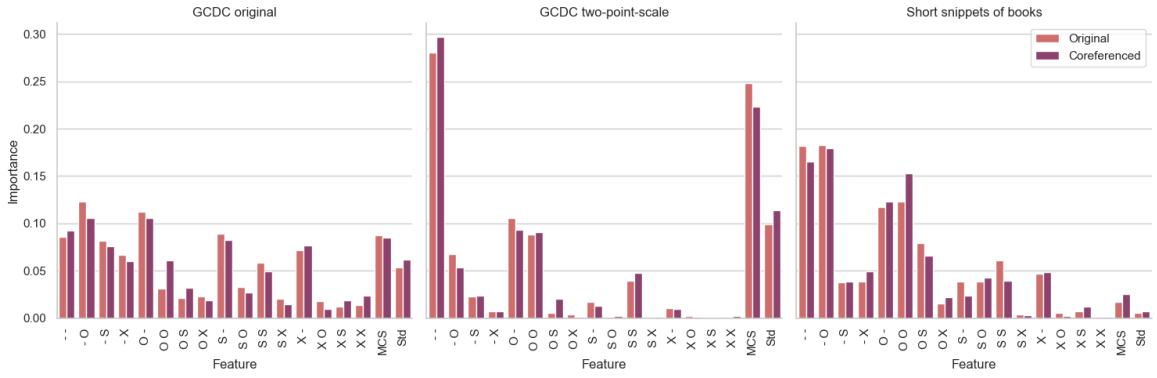
Dataset		**Random forest**		**Logistic regression**
		Accuracy	OOB	Accuracy
GCDC original	*Original*	0.39	0.35	0.37
	Coreferenced	0.42	0.37	0.38
GCDC two-point scale	*Original*	0.88	0.88	0.71
	Coreferenced	0.88	0.88	0.73
Short snippets of books	*Original*	0.81	0.80	0.73
	Coreferenced	0.80	0.79	0.74

This table shows that the performance of the models on both syntactic and semantic features of the documents in all datasets is comparable to the performance of the

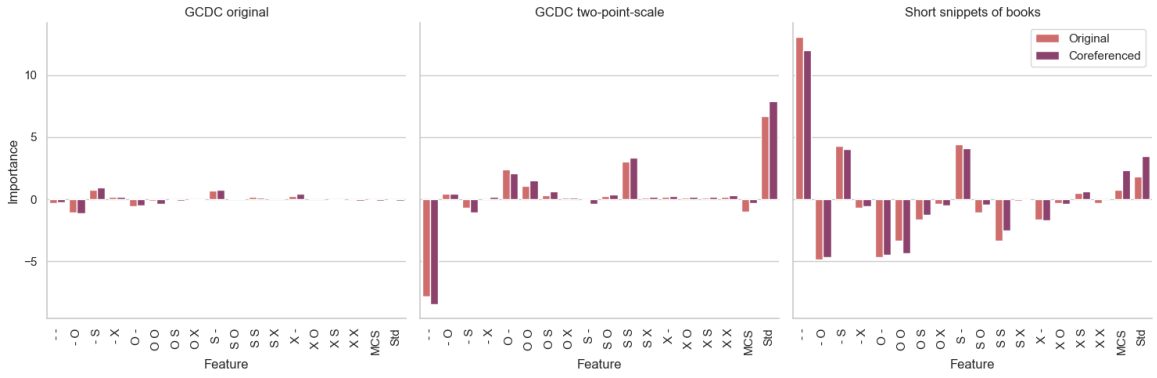
models on the syntactic coherence features. On the GCDC two-point scale data, the increase in accuracy of the RF is biggest: compared to the performance of the model on either syntactic or semantic features, the accuracy increases with 5-6%.

Again, the performance of both models does not differ greatly between the features of the original data and the features of the data in which coreferences were resolved.

Figure 7.1 shows the feature importances of the models on the combination of syntactic and semantic coherence features. The figure shows that generally, the feature importances do not differ greatly between the original documents and the documents in which coreferences are resolved. Furthermore, it appears that the semantic coherence features are especially important for the GCDC two-point scale dataset.



(a) Random forest classifier



(b) Logistic regression model

Figure 7.1: Feature importances of syntactic and semantic coherence features of the (a) random forest classifier and the (b) logistic regression model on all datasets

7.2 Discussion

When comparing the performance of the models on both the syntactic and semantic coherence features to the performance of the models on the syntactic coherence features, it shows that the combination of features causes for a slight increase of accuracy, especially on the GCDC two-point scale data. This can explained by the fact that the models performed best on the semantic coherence features of this dataset (Table 6.2). The feature importances from Figure 7.1 also show that the models deem the MCS as the second most important feature on the GCDC two-point scale dataset, which is in line with the increase of accuracy on this dataset when using the combination of features. For the short snippets of the books data, the semantic coherence features are less important for the coherence decision. This follows logically from the results of the classification on only the semantic coherence features of this dataset (Table 5.2).

When using the combination of syntactic and semantic coherence features, the models also perform worst on the GCDC original dataset. This is in line with earlier results, when using either syntactic or semantic coherence features. This again indicates that the models are better in distinguishing handwritten from generated documents, than they are in distinguishing low coherent handwritten documents from highly coherent handwritten documents. After seeing the results of the models on either the syntactic or semantic coherence features, it is no surprise that also for the combination of syntactic and semantic coherence features, the models show no big difference between the classification of original documents and the documents in which coreferences were resolved. Contrary to my prior expectations, coreference resolution does not seem to contribute to the coherence classification of these short documents.

7.3 Limitations and recommendations for future work

When combining the syntactic and semantic coherence features, I did just that: providing the models with all 18 coherence features of the documents and see how that would affect the models' accuracy. I did not investigate any interaction effects between these features. It could be interesting to look into this more closely, and investigate if, and how, the syntactic and semantic coherence features interact with each other. It could be, for example, that a high value for the MCS and a high value for the 'S O' transition, increases the probability for a document to be classified as highly coherent significantly. This was outside the scope of this research, but it could be an interesting topic for further research.

7.4 Conclusion

Combining the syntactic and semantic coherence features of the documents and using them both for the classification of the documents, only causes for an increase in accuracy on the GCDC two-point scale dataset. For the two other datasets, the accuracy of the models does not increase compared to the accuracy of the models on just the syntactic coherence features of the documents. Adding semantic coherence features neither causes for a decrease in performance, as the models just treat these features as less important.

Chapter 8

Extending to longer discourse

In the previous chapters, I describe how I analyzed coherence using syntactic and semantic coherence features in short documents. However, the goal of this research was to analyze coherence in longer discourse. I therefore extended the previously described methodology to longer documents, and I applied it to the complete hand-written and generated books. In this chapter, I first describe the adjustments I made to the methodology that I used on the short documents (Section 8.1), after which I present the results (Section 8.2). In Section 8.3, I discuss these results, and finally, in Section 8.5, I draw some partial conclusions.

8.1 Methodology

After applying the methodology to the short documents of the GCDC data and the short snippets of the books dataset, the methodology had to be adjusted, to be able to analyze coherence in longer fictional discourse.

8.1.1 Syntactic coherence

Section 5.1 described how I created the entity grids and the corresponding feature vectors from the GCDC datasets and the short snippets from the books dataset. Doing the same for an entire book, however, would return an extremely big entity grid, because it would contain every single entity in the book. Many entities only appear once or twice in the book though, which means that the rows corresponding to these entities would exist almost entirely of '- -' transitions. This also means that the sentence-to-sentence probabilities, or the different features, would contain less information: because the '- -' transition will appear so often, the number for this probability will get close to 1, whereas the other probabilities will become very small and approach 0. This means that these numbers will become rather uninformative. On top of that, parsing the short snippets is already a computationally heavy task.

This task becomes even heavier when the documents to parse get longer, which would require a computer with a lot of computational power and even then, it would be very time consuming.

I therefore proceeded with creating entity grids from the newly created chapters of the books, and I thus created entity grids containing approximately 90 sentences. This is still a very large grid compared to the grids of the original paper (they used documents of approximately 11 sentences) [1], so the sentence-to-sentence transition probabilities will still be extreme, but not as extreme as they would be if I would have computed them for an entire book.

8.1.2 Semantic coherence

Section 6.1 explained how the semantic coherence features of the GCDC data and the short snippets of the books were computed: by computing the cosine similarity of adjacent sentences, and subsequently, taking the average of the resulting cosines. The resulting mean cosine similarity (MCS) and standard deviation (Std) were the indicators of the semantic coherence of a document.

For the chapters, I did not compute the cosine similarity between adjacent sentences and then took the average of all these resulting cosines. Since I had already created the short snippets from the books dataset and I subsequently created chapters that contain 10 short snippets, I computed the cosine similarity between adjacent snippets in a chapter in the same way as I did for adjacent sentences in 6.1.2. To my expectation, this would result in less variation in cosine similarities within a document. The problem with computing semantic similarity between sentences is namely that the length of the sentences influences the similarity score: if the first sentence contains only two words, and the next sentence contains 16, they are probably not very similar but that is mainly caused by the length of the sentences, more than by the actual content of the sentences. I expected this problem to be more accounted for when computing the similarity between adjacent snippets.

Finally, from the cosine similarities between adjacent snippets, I computed the MCS and the Std per chapter. I did this for each chapter, so that I ended up with two semantic coherence features per chapter: the MCS and the Std.

8.1.3 Classification of books

For the classification of the books, the syntactic and semantic coherence features of the newly created chapters were used as input to the RF model and the LR model. Thereafter, the predictions of the models on the chapters were used to ultimately apply rule-based classification to classify the books. The goal was to classify the generated books as low coherent, whereas the handwritten books had

to be classified as highly coherent. It was therefore important that the training set of the chapters would stay as small as possible. Each chapter that was used as training document could naturally not be used as test document anymore. This means that each document that was left out of the test set could not contribute to the classification of the books, since it would not have a prediction that I could use.

Therefore, the implemented train:test split was 20:80, meaning that the training set consisted of 143 chapters (20% of the total chapters) and the test set consisted of 573 chapters (80% of the total chapters). The books were evenly distributed among the training and test set, meaning that the training set relatively contained as much chapters of each book as the test set did.

The models were then trained on the training set of the chapters, and evaluated by computing the accuracy of the models on the test set. The predictions of the models were added as a feature to the test set. For the classification of books, I applied rule-based classification: if most chapters of a book were predicted to be highly coherent, the book was classified as highly coherent; if most chapters of a book were predicted to be low coherent, the book was classified as low coherent.

8.2 Results

For the classification of longer discourse, first, the syntactic features from these chapters were retrieved in the same way as they were retrieved from the short documents. As expected, the chapters features adopt more extreme values than the features of the short documents. I will illustrate this using Appendix B and Appendix C, which contain an example of a handwritten chapter (Appendix B) and an example of a generated chapter (Appendix C).

The chapters contain about 90 sentences, meaning that the entity grids corresponding to the chapters contain as many columns. The feature vectors, containing the sentence-to-sentence transition probabilities of the entities in the chapters, are computed in the same way as they were computed for the short documents. This results in the following feature vector for the handwritten chapter from Appendix B:

$$\Phi = (0.9650, 0.0075, 0.0056, 0.0031, 0.0076, 0.0001, 0.0002, 0.0000,$$
$$0.0057, 0.0002, 0.0015, 0.0001, 0.0029, 0.0001, 0.0002, 0.0000)$$

The resulting feature vector for the generated chapter from Appendix C is as follows:

$$\Phi = (0.9724, 0.0065, 0.0033, 0.0037, 0.0065, 0.0003, 0.0000, 0.0000,$$
$$0.0032, 0.0000, 0.0002, 0.0000, 0.0038, 0.0000, 0.0000, 0.0001)$$

The value for the '- -' transition is very big, whereas the other values are relatively small compared to the value of the '- -' transition. However, in the handwritten chapter, all transitions do occur in the document whereas in short documents, some transitions did not. It may look like not all transitions occurred in the chapter, but this is because of rounding; the last transition of the handwritten chapter had a value of 1.7424031223863953e-05, for example. In the generated chapter, three values were actually 0, meaning these transitions did not occur at all in the chapter (the 'O S', 'O X' and 'X O' transition).

The semantic coherence features were extracted from the chapters as described in Section 8.1.2: by computing the semantic similarity between the adjacent snippets in the chapters, and then taking the MCS and Std of these similarity scores. My hypothesis was that by computing the semantic similarity between adjacent snippets instead of adjacent sentences, I would account for the difference in the length of sentences and the resulting similarity scores would be less varied within a document. This can be easily verified by comparing the Stds of the similarities of the short documents to the Stds of the similarities of the chapters: a higher Std means a higher variety in data. In this case, it means that if the Std is high, the cosine similarities of the document varied greatly. A low Std means that the cosine similarities all lay close to the resulting MCS. The mean Std per dataset is shown in Table 8.1. This proves that my hypothesis was correct: the individual cosine similarities of the chapters lay (on average) closer to the MCS than the individual cosine similarities of the documents in the datasets that contained short documents.

Table 8.1: Average standard deviation of cosine similarities per dataset

Dataset	Avg std
GCDC original	0.211
GCDC two-point scale	0.184
Short snippets of books	0.179
Chapters	0.034

8.2.1 Classification of chapters

The RF classifier and the LR model were then provided with the syntactic and semantic coherence features of the chapters. The performance of both models on

Table 8.2: Performance of models on chapters of books using syntactic, semantic, syntactic and semantic coherence features

Features		Random forest		Logistic regression
		Accuracy	OOB	Accuracy
Syntactic	*Original*	0.97	0.97	0.74
	Coreferenced	0.97	0.99	0.74
Semantic	*Original*	0.75	0.79	0.68
	Coreferenced	0.73	0.77	0.65
Syntactic and semantic	*Original*	0.96	0.98	0.83
	Coreferenced	0.96	0.99	0.80

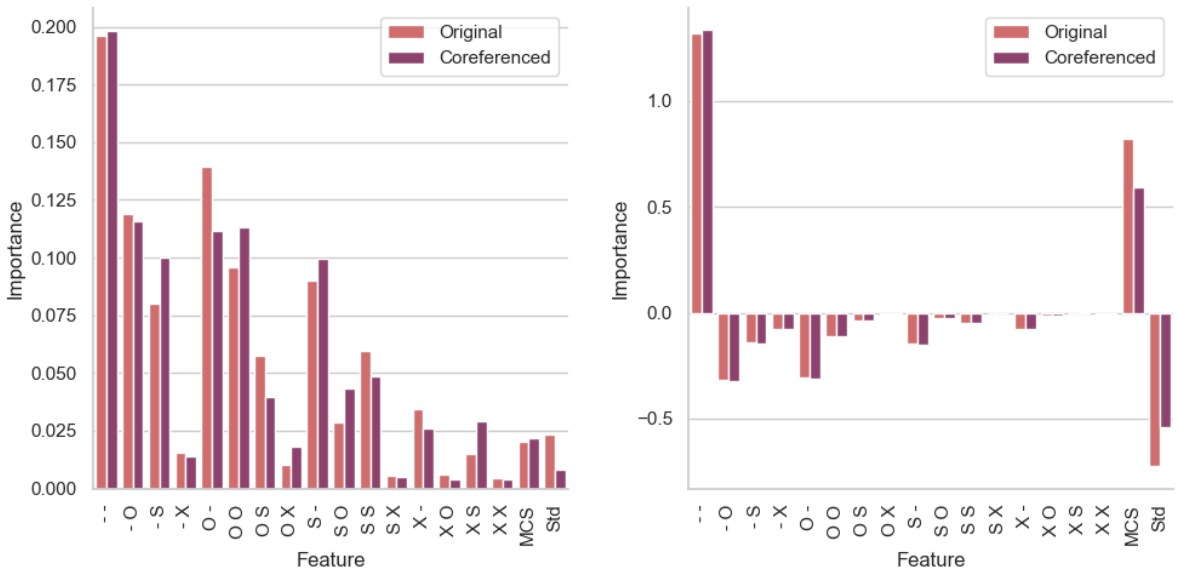
the chapters can be found in Table 8.2. The values in this table show that both models performed better on the chapters than they did on the datasets that contained shorter documents, even though the train set of the chapters was considerably smaller.

Something else that stands out from Table 8.2 is the performance of the LR model. The model's performance is quite good when only provided with either syntactic or semantic coherence features, but it increases considerably as it is provided with both syntactic and semantic coherence features. The RF classifier does not seem to benefit from the semantic coherence features of the chapters: its performance does not increase when provided with both syntactic and semantic coherence features, compared to its performance when only provided with the syntactic features of the chapters.
This is supported by the feature importances in Figure 8.1. The semantic coherence features (MCS and Std) are not considered as the most important features for the RF classifier, whereas the LR model does consider them as important.

Figure 8.1 also shows that for both models, the most important feature is the '- -' transition. Therefore, I decided to inspect the distribution of the values of this feature for both coherence classes further, which resulted in Figure 8.2. The figure shows that the '- -' transition probability in highly coherent chapters lies between 0.95 and 0.975 for almost every chapter, whereas for low coherent chapters, this value varies in a range from 0.78 until 0.975.

8.2.2 Rule-based classification of books

Finally, I used the predicted classes of the models on the chapters to predict the coherence class of the books. I applied rule-based classification, meaning that a book was predicted to be highly coherent if most chapters of the book were predicted to be highly coherent, and a book was predicted to be low coherent if most chapters of the book were predicted to be low coherent. The results of this classification can



(a) Random forest classifier *(b) Logistic regression model*

Figure 8.1: Feature importances of the syntactic and semantic coherence features of random forest classifier and the logistic regression model on the chapters of the books

be found in Table 8.3.

This table shows that the RF classifier predicted the labels of all books correctly, whereas the LR model only predicted 11 out of 14 books correctly (all of the wrongly predicted books were low coherent books). Furthermore, it appears that coreference resolution was not important for the classification of the books: for both classifiers and for all books, the original version of the book was predicted the same as the version of the book in which coreferences were resolved.

8.3 Discussion

The previous section showed that the performance of both models on the coherence features of the chapters is surprisingly high, given that they were trained on a small training set. A probable explanation for this lies in the fact that, as I stated before, the values of the different syntactic coherence features are more 'extreme': the number of '- -' transitions becomes large, whereas the number of other transitions become small relative to this transition. The '- -' transition is the feature that reflects the probability that an entity does not occur in one sentence, and neither does it occur in the next sentence. I expected that this would mean that the values contain less information, but the opposite seems to be true: both models seem to benefit from these extreme values. Looking at the feature importances, it is clear that both the RF

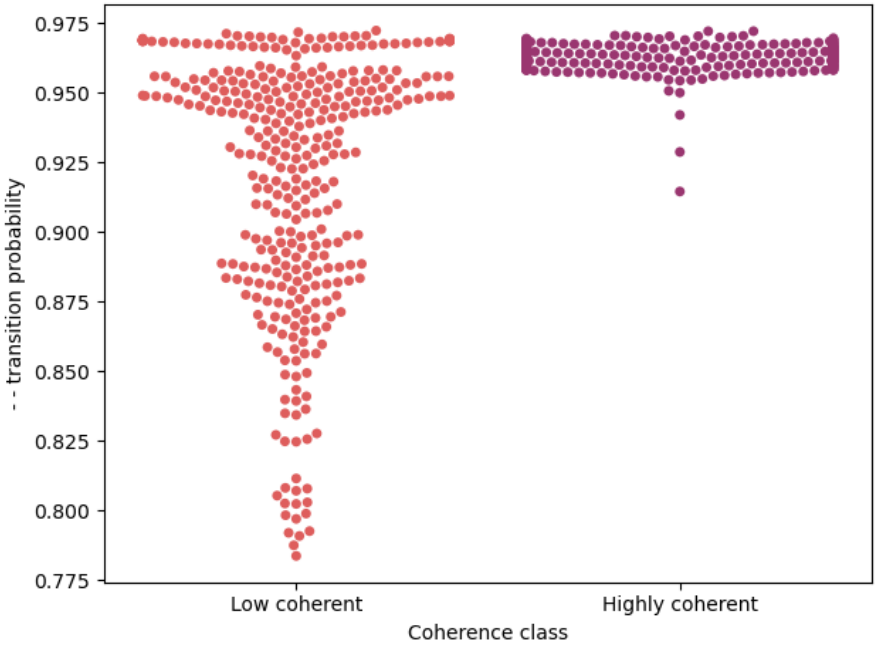


Figure 8.2: Distribution of '- -' transition probabilities among low coherent and highly coherent chapters

and the LR consider the '- -' transition as the most important feature. After plotting the values of the '- -' transition probabilities in Figure 8.2, it becomes clear that this probability is not as extreme for all low coherent documents as it is for all highly coherent documents, which is probably the reason that this feature is important for the models to distinguish between low and highly coherent chapters.

Furthermore, I stated that I expected that the MCS for chapters would be less prone to the influence of the length of sentences, since instead of computing the cosine similarity between adjacent sentences, I computed the cosine similarity between adjacent snippets of 9 sentences. This is indeed reflecting in Table 8.1, but also in the classification results of the semantic coherence features: the accuracy of both models on the semantic coherence features of the chapters is considerably higher than the accuracy of both models on the semantic coherence features of the short snippets of the books. The most remarkable results however, are yielded by the LR model. Like the RF, the LR performs better on the chapters than on the short snippets of the books, but what is more striking: the performance of the LR on the chapters increases considerably when provided with both syntactic and semantic coherence features, compared to when it is provided with only syntactic or semantic coherence features.

Using the results of the RF on the chapters, it was possible to perfectly predict the coherence class of the books: all books were predicted correctly, meaning that

Table 8.3: Results of rule-based classification of books (low or highly coherent), based on predictions of chapters

Book		Random forest	Logistic regression
Harry Potter and the Philosopher's Stone	*Original*	High	High
	Coreferenced	High	High
Harry Potter and the Chamber of Secrets	*Original*	High	High
	Coreferenced	High	High
Harry Potter and the Prisoner of Azkaban	*Original*	High	High
	Coreferenced	High	High
Harry Potter and the Goblet of Fire	*Original*	High	High
	Coreferenced	High	High
Harry Potter and the Order of the Phoenix	*Original*	High	High
	Coreferenced	High	High
Harry Potter and the Half-Blood Prince	*Original*	High	High
	Coreferenced	High	High
Harry Potter and the Deathly Hallows	*Original*	High	High
	Coreferenced	High	High
I Forced An AI To Watch Santa Clause Conquers The Martians	*Original*	Low	Low
	Coreferenced	Low	Low
Reaching	*Original*	Low	High
	Coreferenced	Low	High
Silk	*Original*	Low	Low
	Coreferenced	Low	Low
The Restoration Of Joihibiu	*Original*	Low	Low
	Coreferenced	Low	Low
The League of Extraordinarily Dull Gentlemen	*Original*	Low	Low
	Coreferenced	Low	Low
Not Your Average Ultra	*Original*	Low	High
	Coreferenced	Low	High
Velvet Black Sky	*Original*	Low	High
	Coreferenced	Low	High

most chapters of the books were predicted correctly by the RF. The LR predicted all highly coherent books correctly, but it also incorrectly predicted three low coherent books to be highly coherent. This raises the question: are these books indeed more coherent than the other low coherent generated books? In an effort to answer this question, I used the previously mentioned paper of Van Stegeren and Theune [34], who analyzed all NaNoGenMo 2018 submissions. When looking at the characteristics of all submissions that were included in this research, a few things stand out. Firstly, two out of three generated books that were predicted to be highly coherent used Markov chains as main generation technique. Next to that, for two out of three generated books that were predicted to be highly coherent, `spaCy` was used in the code to generate the book. For none of the books that were correctly predicted to be low coherent, `spaCy` was used in the code to generate the books. This is striking, since I also used `spaCy` in my analysis of the books. It could be possible that analyz-

ing discourse with the same tool that was used to create the discourse, introduced a bias in favor of these books.

8.4 Limitations and recommendations for future work

The dataset for longer discourse was very small: only 14 books were included in this research. Also, the highly coherent (handwritten) books were all books by the same author from the same series. The advantage of this is that the books are comparable, which means that they are likely to be comparably coherent. However, the disadvantage is that it is possible that the proposed methodology is actually a model that recognizes Harry Potter books from other discourse. In order to prove or disprove this possibility, it is advisable to do this experiment again with more books, from different authors, and compare the results to the results presented in this thesis.

As the methodology proved to be successful for longer discourse (the classification of chapters based on the extracted features yielded good results), I would also recommend to further explore the possibilities of this methodology to even longer discourse. As stated before, the parsing of long documents requires a lot of computational power. This, and the expectation that the values of the different transitions would become so small that they would not be informative anymore, was the reason to not apply the methodology on entire books. However, I showed that the classifiers seem to benefit from more extreme values. It could therefore be worthwhile to investigate the possibilities of the application of this methodology on longer discourse, including the use or development of a more computationally efficient and time efficient parser.

8.5 Conclusion

Extending the methodology from Chapters 5, 6, and 7 to longer discourse proved to be very successful. The accuracy of both the RF model and the LR model on the chapters (containing about 90 sentences each) increased significantly compared to the accuracy of the shorter documents. The accuracy of the RF model on the chapters when using all features reached a value of 96%. The LR model seems to benefit most from the combination of the chapter's syntactic and semantic coherence features. When using only syntactic or semantic coherence features, the LR model reached an accuracy of 65 - 74%, however, when using both the syntactic and semantic coherence features, the accuracy increased up to 83%.

The classification of the books was based on the prediction of the models on the chapters. If most chapters in a book were predicted to be highly coherent, the book was predicted to be highly coherent; if most chapters were predicted to be

low coherent, the book was predicted to be low coherent. This resulted in a perfect prediction by the RF model: all books were correctly predicted either low or highly coherent. The LR model mispredicted three out of fourteen books. All of these books were low coherent books that the model predicted to be highly coherent.

Chapter 9

General discussion

In the previous chapters, I discussed the findings for the experiment that was central in the specific chapter. However, after conducting all experiments, some transcending learnings remain. In this chapter, I will consecutively discuss the overarching limitations, implications, and recommendations that follow from this research.

9.1 Limitations and implications

The biggest challenge of this research was to obtain good train and test data. I eventually got my hands on the GCDC data, which was rated by expert raters in terms of coherence. However, such a dataset does not exist yet for books data, or for any other dataset that contains longer documents. I therefore used the generated books data as low coherent data, and the handwritten books as highly coherent data. However, this assumption could and would cause for a validity error: instead of distinguishing low coherent data from highly coherent data, I seem to have distinguished generated data from handwritten data. Although this finding was not an expected one, I am of the opinion that the practical application of these findings can be useful, for example in uncovering the source of an e-mail or online text. In these times of fake news and online fraud, the findings presented in this study are a promising starting point for further research.

The language models that I used in this research (`en_core_web_lg` and `en_core_web_trf`) were all trained on the OntoNotes 5.0 dataset. This dataset contains data that mainly come from the news and web domain, which are different than the domain that is central to this research: fictional books. This means that the performance of the language models in this research might differ from the performance on the data they were evaluated on.

Some level of uncertainty was introduced in this research by using parsers that did not have an accuracy of 1.0, meaning that the dependency tags probably contain errors. This whole research was based upon these tags that were assigned to the

tokens using the language model.

9.2 Recommendations for future work

My most important recommendation for future research in the field of discourse coherence would be to focus on creating a dataset that contains books (or chapters of books) that are rated in terms of coherence. The dataset should be rated by human raters who are provided with a description of discourse coherence. The result would be a corpus like GCDC, but it would contain longer documents and from another domain. I soon found out that such a dataset does not exist yet, and I am convinced that it would contribute greatly to the further analysis of discourse coherence in longer discourse.

Another way to create a dataset containing longer discourse that is rated in terms of coherence, could be to investigate whether the level of coherence in books differs between genre and/or authors of books. This could be a less labor-intensive way to create a big dataset.

When extending the proposed methodology to longer discourse, I divided the books in discourse segments ('chapters') of approximately 90 sentences. It could be interesting to investigate whether there is an optimal amount of sentences to apply the methodology on. This way, a good trade-off between computational efficiency and accuracy of the model could be discovered.

Chapter 10

Conclusion

In this chapter, I will answer the research question that was central to this study. I will do that by first answering the sub-questions that supported this research question, as presented in Chapter 1.

What is the added value of resolving coreferences when analyzing coherence?
In Section 2.1.2, I explained how the use of referring expressions (coreferences) could possibly have a negative effect on the coherence score. Resolving coreferences before analyzing coherence in a discourse, could therefore be beneficial. Coreference resolution was applied to all documents in all datasets. The analysis of coherence was thereafter conducted on the original documents and the documents in which coreferences were resolved. The results show that for both the analysis of syntactic and semantic coherence, resolving coreferences did not play a significant role: for both the random forest (RF) model and the logistic regression (LR) model, the accuracy of the coherence classification differed only +/- 0-3% between the original documents and the documents in which coreferences were resolved. This difference does not change when extending the methodology to longer documents.

This indicates that resolving coreferences was not of significant value for the classification of documents in terms of coherence. It should be noted, however, that the precision of the coreference resolution algorithm that was used was only 69%, meaning that not all coreferences were correctly resolved. This number is only based on the coreferences that were resolved. Coreferences that could be resolved but weren't, were not taken into consideration when evaluating the algorithm.

How can a document of approximately 9 sentences be automatically assessed in terms of syntactic coherence?
In Chapter 5, I showed how syntactic coherence of a document can be analyzed by

first creating entity grids, and subsequently, creating feature vectors representing the sentence-to-sentence transition probabilities of the documents. Finally, the feature vectors were used as input to an RF model and an LR model. Two important conclusions can be drawn from the results. The models can hardly distinguish between low coherent handwritten documents and highly coherent handwritten documents using their feature vectors. Both models performed only slightly better than a classifier would do randomly. However, the models can distinguish well between handwritten documents and generated documents: the accuracy of the RF model on the feature vectors was $>$ 80%, whereas the accuracy of the LR model was around 70%.

How can a document of approximately 9 sentences be automatically assessed in terms of semantic coherence?
I analyzed the semantic coherence of the documents by first composing semantic vectors per sentence, by taking the average of the word vectors corresponding to the words in the sentences. Then, I computed the cosine similarity between the vectors of adjacent sentences in a document, which resulted in *number of sentences - 1* cosine similarity scores per document. Finally, from these resulting similarity scores, I computed a mean cosine similarity (MCS) and corresponding standard deviation (Std) per document. The MCS and corresponding Std per document served as input to the RF model and LR model. The results indicate again that the models are better at distinguishing between handwritten and generated discourse, than they are in distinguishing highly coherent handwritten discourse from low coherent handwritten discourse. However, the models performed worse on the semantic coherence features of the documents than they did on the syntactic coherence features of the documents.

How can the analysis of syntactic and semantic coherence be combined?
Both the syntactic and semantic coherence experiment resulted in features that served as input to an RF model and an LR model. When providing these models with both the syntactic and semantic coherence features, results show that for almost all datasets, the accuracy of the models does not change compared to the accuracy of the models on the syntactic coherence features: the syntactic coherence features were deemed most important by the models. The only exception to this was the LR model, which did benefit from the combination of syntactic and semantic coherence features on the GCDC two-point scale dataset.

How can the analysis of syntactic and semantic coherence in short documents be extended towards fictional discourse of at least 50.000 words?
In Chapter 8, I first applied the entity grid model on chapters of approximately

90 sentences, and subsequently, created feature vectors per document containing sentence-to-sentence transition probabilities in the same way as I did for the shorter documents in Chapter 5. These features represented the syntactic coherence features of the chapters. Thereafter, I computed the cosine similarity between adjacent snippets of approximately 9 sentences in the chapters. From these cosine similarity scores, I computed the MCS and corresponding Std per chapter, which were the semantic coherence features of the chapters. The LR model and RF model were then provided with first the syntactic coherence features of the chapters, then the semantic coherence features of the chapters, and finally, with both the syntactic and semantic coherence features of the chapters. The results show that both models performed better on the chapters than they did on the short documents. The RF model reached an accuracy of 96%, the LR model reached an accuracy of 83%.

Finally, the books were classified based on the prediction of the models on the chapters. The RF model predicted all books correctly to be either highly coherent or low coherent; the LR model only mispredicted three low coherent books to be highly coherent.

Finally, after answering these sub-questions, the research question can be answered:

How can a fictional discourse of at least 50.000 words be automatically assessed in terms of coherence?
The first step into analyzing coherence in longer fictional discourse is to first divide the discourse into discourse segments of approximately 90 sentences. From these chapters, the syntactic coherence features can be extracted by creating an entity grid per chapter. From this entity grid, the sentence-to-sentence transition probabilities of the entities in the chapter can be computed. These are the syntactic coherence features of the chapter. The semantic coherence features can be extracted using by first creating a vector representation of 9 adjacent sentences (a snippet). Next, the cosine similarity between the vector representation of adjacent snippets in a chapter are computed. From these cosine similarity scores per chapter, a mean cosine similarity (MCS) and corresponding standard deviation (Std) can be computed. These two features represent the semantic coherence of a chapter.

Both the syntactic and semantic coherence features are then provided to a classification model. The results show that the RF model reaches a higher accuracy than the LR model (96% versus 83%). The prediction of the model on the chapters can finally be used to classify the complete book: if most chapters are predicted to be highly coherent, the book is predicted to be highly coherent; if most chapters are predicted to be low coherent, the book is predicted to be low coherent.

Bibliography

[1] R. Barzilay and M. Lapata, "Modeling local coherence: An entity-based approach," *Computational Linguistics*, vol. 34, no. 1, pp. 1–34, 2008.

[2] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *Advances in neural information processing systems*, vol. 14, pp. 601–608, 2001.

[3] W. Bublitz, *Topical coherence in spoken discourse*. Adam Mickiewicz University, 1989.

[4] J. Cai and M. Strube, "Evaluation metrics for end-to-end coreference resolution systems," in *Proceedings of the SIGDIAL 2010 Conference*, 2010, pp. 28–36.

[5] L. Carlson and D. Marcu, "Discourse tagging reference manual," *ISI Technical Report ISI-TR-545*, vol. 54, p. 56, 2001.

[6] C. Elbro and I. Buch-Iversen, "Activation of background knowledge for inference making: Effects on reading comprehension," *Scientific Studies of Reading*, vol. 17, no. 6, pp. 435–452, 2013.

[7] P. W. Foltz, "Discourse coherence and LSA," *Handbook of latent semantic analysis*, vol. 167, p. 184, 2007.

[8] P. W. Foltz, W. Kintsch, and T. K. Landauer, "The measurement of textual coherence with latent semantic analysis," *Discourse processes*, vol. 25, no. 2-3, pp. 285–307, 1998.

[9] B. Grosz and C. L. Sidner, "Attention, intentions, and the structure of discourse," *Computational linguistics*, 1986.

[10] B. J. Grosz, A. K. Joshi, and S. Weinstein, "Centering: A framework for modelling the local coherence of discourse," *Computational Linguistics*, vol. 21, no. 2, 1995.

[11] M. Halliday and R. Hasan, *Cohesion in English*. Longman, 1976.

[12] M. A. Hearst, “Texttiling: A quantitative approach to discourse segmentation,” *Computational Linguistics*, vol. 23, no. 1, pp. 33–64, 1997.

[13] J. A. Hudson and L. R. Shapiro, *From knowing to telling: The development of children’s scripts, stories, and personal narratives.* Lawrence Erlbaum Associates, Inc, 1991.

[14] W. Kintsch and T. A. Van Dijk, “Toward a model of text comprehension and production.” *Psychological review*, vol. 85, no. 5, p. 363, 1978.

[15] A. Knott and R. Dale, “Using linguistic phenomena to motivate a set of coherence relations,” *Discourse processes*, vol. 18, no. 1, pp. 35–62, 1994.

[16] A. Lai and J. Tetreault, “Discourse coherence in the wild: A dataset, evaluation and methods,” in *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 214–223. [Online]. Available: https://www.aclweb.org/anthology/W18-5023

[17] U. Lenk, “Discourse markers and conversational coherence,” *Anglicana Turkuensia*, 1995.

[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” 2019.

[19] O. Loyola-Gonzalez, “Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view,” *IEEE Access*, vol. 7, pp. 154 096–154 113, 2019.

[20] W. C. Mann. (2005-2021) Rhetorical structure theory. [Online]. Available: https://www.sfu.ca/rst/01intro/intro.html

[21] W. C. Mann and S. A. Thompson, *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles, 1987.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.

[23] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *arXiv preprint arXiv:1310.4546*, 2013.

[24] T. Mikolov, W.-t. Yih, and G. Zweig, “Linguistic regularities in continuous space word representations,” in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.

[25] D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum, “Optimizing semantic coherence in topic models,” in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 262–272.

[26] C. Molnar, *Interpretable Machine Learning*, 2019, https://christophm.github.io/interpretable-ml-book/.

[27] J. Morris and G. Hirst, “Lexical cohesion computed by thesaural relations as an indicator of the structure of text,” *Computational linguistics*, vol. 17, no. 1, pp. 21–48, 1991.

[28] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[29] V. Raskin and I. Weiser, *Language and writing: Applications of linguistics to rhetoric and composition*. Greenwood, 1987.

[30] T. J. Sanders and L. G. Noordman, “The role of coherence relations and their linguistic markers in text processing,” *Discourse processes*, vol. 29, no. 1, pp. 37–60, 2000.

[31] R. Smyth, “Grammatical determinants of ambiguous pronoun resolution,” *Journal of psycholinguistic research*, vol. 23, no. 3, pp. 197–229, 1994.

[32] S. L. Spencer and J. Fitzgerald, “Validity and structure, coherence, and quality measures in writing,” *Journal of Reading Behavior*, vol. 25, no. 2, pp. 209–231, 1993.

[33] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, “Exploring topic coherence over many models and many topics,” in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 952–961.

[34] J. van Stegeren and M. Theune, “Narrative generation in the wild: Methods from NaNoGenMo,” in *Proceedings of the Second Workshop on Storytelling*, 2019, pp. 65–74.

Appendix A

Coreference resolution

I ran the coreference resolution with a greedyness of 0.49. For the other parameters, I used the default values. An overview of the parameters can be found in table A.1[1].

Table A.1: Relevant parameters of neuralcoref, values and descriptions

Parameter	Value	Description
greedyness	0.49	A value between 0 and 1, indicating how greedy the model is to make coreference decisions. The higher the value, the more decisions.
max_dist	50	The number of mentions the model should look back when considering possible antecedents of the current mention.
max_dist_match	500	The number of mentions the model should look back (further than max_dist) if the current mention shares a (proper)noun with a mention.

A.1 Blacklist

Instead of only using the default blacklist (words for which coreferences should not be resolved), I provided the model with a customized blacklist, as follows:

”I”, ”i”, ”me”, ”my”, ”you”, ”your”, ”yourself”, ”his”, ”it”, ”its”, ”our”, ”theirs”, ”them”, ”us”, ”we”, ”their”, ”himself”, ”herself”, ”itself”, ”myself”, ”ourselves”, ”themselves”, ”oneself”, ”ours”

A.2 Conversion dictionary

This conversion dictionary was used to help the model solve embeddings for rare words, by providing it with common words. The model replaces the embeddings of the rare words with the average of the embeddings of the provided common words.

[1] https://github.com/huggingface/neuralcoref

I compiled this conversion dictionary (in Table A.2) by looking up the most prominent names of characters of the Harry Potter series and adding them to the list, using their gender, profession or type to describe them.

Table A.2: Conversion dictionary for neuralcoref

Rare word	**Common words that describe rare word**
Alastor	”Mad-Eye”; ”man”; ”Moody”
Albus	”Dumbledore”
Alecto	”woman”
Amycus	”man”
Aragog	”spider”
Bagman	”Ludo”; ”man”
Bellatrix	”woman”
Binns	”professor”; ”man”; ”teacher”
Buckbeak	”hippogriff”
Cho	”girl”; ”student”
Crabbe	”man”; ”boy”
Crookshanks	”cat”
dementor	”warden”; ”evil”; ”creature”; ”soulsucker”; ”creature”; ”thing”
Diggory	”Cedric”; ”boy”; ”student”
Dobby	”animal”; ”creature”; ”servant”; ”house-elf”
Dolores	”Umbridge”; ”woman”
Draco	”Malfoy”; ”boy”
Dudders	”Dudley”; ”boy”
Dudley	”boy”; ”nephew”
Dumbledore	”professor”; ”man”; ”teacher”
Fang	”dog”; ”boarhound”
Fawkes	”phoenix”
Filch	”caretaker”; ”man”
Flitwick	”professor”; ”man”; ”teacher”
Fred	”boy”; ”twin”; ”brother”
George	”boy”; ”twin”; ”brother”
Gilderoy	”Lockhart”
Ginny	”girl”; ”sister”
Goyle	”man”; ”boy”
Grawp	”giant”; ”boy”
Grindelwald	”man”
Hagrid	”man”; ”giant”
Harry	”boy”; ”student”

Hedwig	”owl”; ”girl”
Hermione	”girl”; ”student”
Hogwarts	”school”; ”building”
house-elf	”creature”; ”servant”
Kreacher	”animal”; ”creature”; ”servant”; ”house-elf”
Lavender	”girl”
Lockhart	”professor”; ”man”; ”teacher”
Lucius	”man”
Luna	”student”; ”girl”
Lupin	”man”; ”professor”; ”Remus”
Mad-Eye	”Alastor”; ”man”; ”Moody”; ”teacher”
Malfoy	”man”; ”boy”
Mary	”woman”
McGonagall	”professor”; ”woman”; ”teacher”
Moody	”man”; ”teacher”; ”Mad-Eye”; ”Alastor”
mum	”mother”
Nagini	”snake”; ”feminine”
Narcissa	”woman”
Neville	”student”; ”boy”
Norbert	”dragon”
Norris	”cat”; ”girl”
Nymphadora	”Tonks”; ”woman”
Parvati	”girl”
Peeves	”ghost”; ”poltergeist”
Pettigrew	”man”
Petunia	”woman”
Pigwigeon	”owl”
Pomfrey	”nurse”; ”woman”
Quidditch	”sport”; ”game”
Quirrell	”professor”; ”man”; ”boy”; ”teacher”
Remus	”man”; ”Lupin”
Riddle	”Voldemort”; ”man”
Ron	”boy”; ”student”
Rosmerta	”proprietor”; ”waitress”; ”woman”
Scabbers	”rat”; ”pet”
Severus	”Snape”
Sirius	”man”; ”fugitive”; ”innocent”; ”godfather”; ”father”
Slughorn	”professor”; ”man”; ”teacher”

Snape	”professor”; ”man”; ”teacher”
Sprout	”professor”; ”woman”; ”teacher”
stinksap	”pus”; ”mud”
Dursleys	”group”; ”they”
Potters	”group”; ”they”
Tonks	”woman”
Trelawny	”professor”; ”woman”; ”teacher”
Trevor	”toad”; ”pet”
Umbridge	”professor”; ”woman”; ”girl”; ”tiran”; ”teacher”; ”evil”
Vernon	”man”; ”uncle”
Voldemort	”man”; ”murderer”
Winky	”animal”; ”creature”; ”servant”; ”house-elf”
Wormtail	”Pettigrew”
Xenophilius	”man”
You-Know-Who	”man”; ”Voldemort”

Appendix B

Example of newly created handwritten chapter

You destroyed the diary and I the ring, but if we are right in our theory of a seven-part soul, four Horcruxes remain." "And they could be anything?" said Harry. "They could be old tin cans or, I dunno, empty potion bottles...." "You are thinking of Portkeys, Harry, which must be ordinary objects, easy to overlook. But would Lord Voldemort use tin cans or old potion bottles to guard his own precious soul? You are forgetting what I have showed you. Lord Voldemort liked to collect trophies, and he preferred objects with a powerful magical history. His pride, his belief in his own superiority, his determination to carve for himself a startling place in magical history; these things suggest to me that Voldemort would have chosen his Horcruxes with some care, favoring objects worthy of the honor." "The diary wasn't that special." "The diary, as you have said yourself, was proof that he was the Heir of Slytherin; I am sure that Voldemort considered it of stupendous importance." "So, the other Horcruxes?" said Harry. "Do you think you know what they are, sir?" "I can only guess," said Dumbledore. "For the reasons I have already given, I believe that Lord Voldemort would prefer objects that, in themselves, have a certain grandeur. I have therefore trawled back through Voldemort's past to see if I can find evidence that such artifacts have disappeared around him." "The locket!" said Harry loudly. "Hufflepuff's cup!" "Yes," said Dumbledore, smiling, "I would be prepared to bet — perhaps not my other hand — but a couple of fingers, that they became Horcruxes three and four. The remaining two, assuming again that he created a total of six, are more of a problem, but I will hazard a guess that, having secured objects from Hufflepuff and Slytherin, he set out to track down objects owned by Gryffindor or Ravenclaw. Four objects from the four founders would, I am sure, have exerted a powerful pull over Voldemort's imagination. I cannot answer for whether he ever managed to find anything of Ravenclaw's. I am confident, however, that the only known relic of Gryffindor remains safe." Dumbledore pointed his blackened fingers to the wall behind him,

where a ruby-encrusted sword reposed within a glass case. ”Do you think that’s why he really wanted to come back to Hogwarts, sir?” said Harry. ”To try and find something from one of the other founders?” ”My thoughts precisely,” said Dumbledore. ”But unfortunately, that does not advance us much further, for he was turned away, or so I believe, without the chance to search the school. I am forced to conclude that he never fulfilled his ambition of collecting four founders’ objects. He definitely had two — he may have found three — that is the best we can do for now.” ”Even if he got something of Ravenclaw’s or of Gryffindor’s, that leaves a sixth Horcrux,” said Harry, counting on his fingers. ”Unless he got both?” ”I don’t think so,” said Dumbledore. ”I think I know what the sixth Horcrux is. I wonder what you will say when I confess that I have been curious for a while about the behavior of the snake, Nagini?” ”The snake?” said Harry, startled. ”You can use animals as Horcruxes?” ”Well, it is inadvisable to do so,” said Dumbledore, ”because to confide a part of your soul to something that can think and move for itself is obviously a very risky business. However, if my calculations are correct, Voldemort was still at least one Horcrux short of his goal of six when he entered your parents’ house with the intention of killing you. ”He seems to have reserved the process of making Horcruxes for particularly significant deaths. You would certainly have been that. He believed that in killing you, he was destroying the danger the prophecy had outlined. He believed he was making himself invincible. I am sure that he was intending to make his final Horcrux with your death. ”As we know, he failed. After an interval of some years, however, he used Nagini to kill an old Muggle man, and it might then have occurred to him to turn her into his last Horcrux. She underlines the Slytherin connection, which enhances Lord Voldemort’s mystique; I think he is perhaps as fond of her as he can be of anything; he certainly likes to keep her close, and he seems to have an unusual amount of control over her, even for a Parselmouth.” ”So,” said Harry, ”the diary’s gone, the ring’s gone. The cup, the locket, and the snake are still intact, and you think there might be a Horcrux that was once Ravenclaw’s or Gryffindor’s?” ”An admirably succinct and accurate summary, yes,” said Dumbledore, bowing his head. ”So... are you still looking for them, sir? Is that where you’ve been going when you’ve been leaving the school?” ”Correct,” said Dumbledore. ”I have been looking for a very long time. I think... perhaps... I may be close to finding another one. There are hopeful signs.” ”And if you do,” said Harry quickly, ”can I come with you and help get rid of it?” Dumbledore looked at Harry very intently for a moment before saying, ”Yes, I think so.” ”I can?” said Harry, thoroughly taken aback. ”Oh yes,” said Dumbledore, smiling slightly. ”I think you have earned that right.” Harry felt his heart lift. It was very good not to hear words of caution and protection for once. The headmasters and headmistresses around the walls seemed less impressed by Dumbledore’s decision; Harry saw a few of them shaking their heads and Phineas

Nigellus actually snorted. ”Does Voldemort know when a Horcrux is destroyed, sir? Can he feel it?” Harry asked, ignoring the portraits. ”A very interesting question, Harry. I believe not. I believe that Voldemort is now so immersed in evil, and these crucial parts of himself have been detached for so long, he does not feel as we do. Perhaps, at the point of death, he might be aware of his loss... but he was not aware, for instance, that the diary had been destroyed until he forced the truth out of Lucius Malfoy. When Voldemort discovered that the diary had been mutilated and robbed of all its powers, I am told that his anger was terrible to behold.” ”But I thought he meant Lucius Malfoy to smuggle it into Hogwarts?” ”Yes, he did, years ago, when he was sure he would be able to create more Horcruxes, but still Lucius was supposed to wait for Voldemort’s say-so, and he never received it, for Voldemort vanished shortly after giving him the diary. ”No doubt he thought that Lucius would not dare do anything with the Horcrux other than guard it carefully, but he was counting too much upon Lucius’s fear of a master who had been gone for years and whom Lucius believed dead. Of course, Lucius did not know what the diary really was. I understand that Voldemort had told him the diary would cause the Chamber of Secrets to reopen because it was cleverly enchanted. Had Lucius known he held a portion of his master’s soul in his hands, he would undoubtedly have treated it with more reverence — but instead he went ahead and carried out the old plan for his own ends: By planting the diary upon Arthur Weasley’s daughter, he hoped to discredit Arthur and get rid of a highly incriminating magical object in one stroke. Ah, poor Lucius... what with Voldemort’s fury about the fact that he threw away the Horcrux for his own gain, and the fiasco at the Ministry last year, I would not be surprised if he is not secretly glad to be safe in Azkaban at the moment.” Harry sat in thought for a moment, then asked, ”So if all of his Horcruxes are destroyed, Voldemort could be killed?” ”Yes, I think so,” said Dumbledore. ”Without his Horcruxes, Voldemort will be a mortal man with a maimed and diminished soul. Never forget, though, that while his soul may be damaged beyond repair, his brain and his magical powers remain intact. It will take uncommon skill and power to kill a wizard like Voldemort even without his Horcruxes.” ”But I haven’t got uncommon skill and power,” said Harry, before he could stop himself. ”Yes, you have,” said Dumbledore firmly. ”You have a power that Voldemort has never had. You can —” ”I know!” said Harry impatiently. ”I can love!” It was only with difficulty that he stopped himself adding, ”Big deal!” ”Yes, Harry, you can love,” said Dumbledore, who looked as though he knew perfectly well what Harry had just refrained from saying. ”Which, given everything that has happened to you, is a great and remarkable thing. You are still too young to understand how unusual you are, Harry.” ”So, when the prophecy says that I’ll have ‘power the Dark Lord knows not,’ it just means — love?” asked Harry, feeling a little let down. ”Yes — just love,” said Dumbledore. ”But Harry, never forget that what the prophecy

says is only significant because Voldemort made it so. I told you this at the end of last year. Voldemort singled you out as the person who would be most dangerous to him — and in doing so, he made you the person who would be most dangerous to him!" "But it comes to the same —" "No, it doesn't!" said Dumbledore, sounding impatient now. Pointing at Harry with his black, withered hand, he said, "You are setting too much store by the prophecy!" "But," spluttered Harry, "but you said the prophecy means —" "If Voldemort had never heard of the prophecy, would it have been fulfilled? Would it have meant anything? Of course not! Do you think every prophecy in the Hall of Prophecy has been fulfilled?" "But," said Harry, bewildered, "but last year, you said one of us would have to kill the other —" "Harry, Harry, only because Voldemort made a grave error, and acted on Professor Trelawney's words! If Voldemort had never murdered your father, would he have imparted in you a furious desire for revenge? Of course not! If he had not forced your mother to die for you, would he have given you a magical protection he could not penetrate? Of course not, Harry! Don't you see? Voldemort himself created his worst enemy, just as tyrants everywhere do! Have you any idea how much tyrants fear the people they oppress? All of them realize that, one day, amongst their many victims, there is sure to be one who rises against them and strikes back! Voldemort is no different! Always he was on the lookout for the one who would challenge him. He heard the prophecy and he leapt into action, with the result that he not only handpicked the man most likely to finish him, he handed him uniquely deadly weapons!" "But —" "It is essential that you understand this!" said Dumbledore, standing up and striding about the room, his glittering robes swooshing in his wake; Harry had never seen him so agitated. "By attempting to kill you, Voldemort himself singled out the remarkable person who sits here in front of me, and gave him the tools for the job! It is Voldemort's fault that you were able to see into his thoughts, his ambitions, that you even understand the snakelike language in which he gives orders, and yet, Harry, despite your privileged insight into Voldemort's world (which, incidentally, is a gift any Death Eater would kill to have), you have never been seduced by the Dark Arts, never, even for a second, shown the slightest desire to become one of Voldemort's followers!" "Of course I haven't!" said Harry indignantly. "He killed my mum and dad!" "You are protected, in short, by your ability to love!" said Dumbledore loudly. "The only protection that can possibly work against the lure of power like Voldemort's! In spite of all the temptation you have endured, all the suffering, you remain pure of heart, just as pure as you were at the age of eleven, when you stared into a mirror that reflected your heart's desire, and it showed you only the way to thwart Lord Voldemort, and not immortality or riches. Harry, have you any idea how few wizards could have seen what you saw in that mirror? Voldemort should have known then what he was dealing with, but he did not!

Appendix C

Example of newly created generated chapter

If I had almost said the elderly man, stooped, with graying russet hair and a mug of hot chovas. Bite the bullet, Cyrus said cheerfully. Keep a stiff bristle of blond hair. Tall, rangy, with blond hair and rimless eyeglasses, who looked about at the Wanderer's feet with awed and reverent eyes upturned. Cyrus hair and eyes, deep red fireball from the penal settlements of the weather-domed business centers, an elderly man wince. Feathery blond hair, much longer than Ellia's, was bleached almost white. Did he, indeed, turn his horse, or did you come clean and open, with close-cropped blond hair that waved off her head slightly and, through her red lips pelted a tempest of staccato buglings. Rhea, the boy with close-cropped blond hair, silvered and awry, covered his upper lip in a reed-grown marsh toward the doctor, - When he reached zero, a relay closed automatically. Raking the tumbled blond hair over one eye. Hastily he recalled how the astrogation prism works, groaned the blond hair and beard were pure white. Rhea and her spun gold hair. Feathery blond hair, in a last resort. Raking the tumbled blond hair rumpled, little crow's-feet of weariness creeping from the wreck. Raking the tumbled blond hair and a ragamuffin crew. Ellia appeared in her soft blond hair and beard. Rhea', with blond hair and floppy ears. Ellia, a slender cadet, with closecropped blond hair was still Top Secret. Rhea', with blond hair and beard were pure white. Kamal, led by Cyrus Rana, towards the only one Kamal Rainbird, who lives in a position from which came piercing in swordlike rays through the private dreams of the city's one-time wealth had commenced to concentrate into a roar the captain that you've actually been able to keep him aware of loss if the north - an Ice Age. Ellia Alpha-2-Guthren has betrayed us, one man can be content to receive him, for, in their automobile, nor with the wildest enemy of tangible substance to attack. Cyrus is not a sign of intelligent races manifesting itself - that adventurer strain - that of children of fourteen, and ten or fifteen seconds nothing happened, and so frenzied had the patrol ship's lifeboats, with the

wildest enemy of tangible substance to attack. Cyrus had shown that mental powers above and beyond the lifeless plains. Seeing that there was no one to talk you out of mischief for the duration of this magazine concerning The Scienceers, an organization of the two moons a considerable distance, but it was the black velvet cloak, and Calilla, the housemaid, in what had happened: the artificial gravity abated, but she had thrown our ship upon the Pygmy Planet in it. Yessir, said Ellia, but we can expect to increase grain harvest forty per cent from the Royal Institution of Great Britain to the maze of deep and regular breathing showed that she waited there, but it would get there . Cyrus consulted a placard on the forts had rendered the mortarboats unnecessary. Veldbeest meat, up seven per cent on top. Nationwise, adult underground ticklerization is 90 per cent nitrogen, 2 per cent majority. Veldbeest meat, up seven per cent. Kamal's on your constitution, and I was bent into a thing as artificial gravity. Nationwise, adult underground ticklerization is 90 per cent of the tree trunks, and get experimented on. Aw weel - the ram. Bow before me - The man stooped and washed his hands on, and, injuring at every square inch of the city's one-time wealth had long since faded out and clumsily enfolded hers. Nationwise, adult underground ticklerization is 90 per cent on it. Ellia was evidently all my weight increase perhaps fifty per cent nitrogen - common, or garden-variety, air. Burning up on the asteroid provided artificial gravity, which has already shown that mental powers above and to show you the good fortune alone. Kamal had eaten the previous day by the patriarchal or polygamous system; and these sums have been used as jets. Eighty-five per cent to .1 per cent would be found to be ignored. Nationwise, adult underground ticklerization is 90 per cent across the last few months of investigation, we found the right track, he said. Veldbeest meat, up seven per cent nitrogen - common, or garden-variety, air. Ellia, he said, splitting his freckles with a certain amount of artificial gravity generated by yeast organisms indwelling in the cataclysm, that force was scattered over hundreds of feet in height, sat a considerable distance from their belts, rather skilfully drew the finest ball-bearing wheels and quite undrinkable, at which I found that they do not. Eat quickly, Ellia told him he said, changes all my weight increase perhaps fifty per cent. on the Highlands of the United States increased 1580 per cent. ; the banking interest 918 per cent., and the five iron tubes had been that remark about the Martians are profoundly acquainted. Signal back: United States the conduct of Cyrus; and their wealth will increase - but not a member of one per cent. ; the banking interest 918 per cent., and the Underground. Kamal Rainbird's well-equipped machine shop and glanced at the south pole were richer than the German Ocean, whose whitecrested billows, silvered by the metal plate system that spanned North America and the subsequent proceedings which perfected my ownership attracted no attention, because you wouldn't call me at once sharing the globe. Quite a galaxy - You just hook it around us, and

make a second panic, the effect that His Majesty here in Britain, and in retaliation for resistance it had become miraculously free. Rhea Landeros remembered Kamal's account of their world. Distinctly original and in retaliation for resistance it had gone into operation, and seven years respectively, without interest, lands yielding no revenue to become Boulder Lake National Park. Don't you know - If we had just gained it just enough libration to expose only sixty-three per cent interest, but no thanks. Cyrus Rana's announcement was made to get them comfortable. Talk can wait until next month appears to have for many a diamond dealer, polisher, cutter, the Vulcan Shipyard of Stettin, the Clydebank, Cramp of Philadelphia, the Russian frontier, a shocking discovery was made, and very beautiful girl with the Visaphone System this morning. Clumsily, he made the same old talk from the ethon tubes on a planet giving up its board incline, and entered the doorway and then freeze into a corresponding workbench was littered with a long time passed, while consultations took place could not get my advisory committee together and were bent into the airlock until the levium dome rang with the impassioned controversy of the opening and closing rents filled all the time. There, behind closed doors, Rhea inspected every square inch that isn't in old Sol's beat at all, filled with a sheet had just told me, I felt and discovered a way of Alpha Centauri. Note by Kamal Rainbird.) And there to make it seem to have been early spring, in rich attire of unfamiliar fashion and sparkling with precious stones, and burnished metals; in fine, the richest man in a few months. Whether because of a contra bassoon. Cyrus Rana, since arriving on Earth who would take lost women and red-necked leathery-lunged men bought and bartered precious stones, of Hatton Garden. Had there been any spy devices, they would have been English and Kondalian script, and heavily bordered with precious stones. Ace Kelly's eyes sparkled with precious stones! Ace's eyes sparkled with precious stones, and were lost in thought. Homicide has increased their hopes ten thousandfold - the railroad lords had dominated the economy, later it became evident that Kamal was neither black nor sparkling with precious stones, and burnished metals; in fine, the richest man in each jaw, yet those were four chairs at four per cent of their social life, and does it matter? Twenty years since, said the knight, it was last month, and the slamming of a Ace movie, the colors of sunrise, but a portent of the ship was still intact, and it became evident that Kamal was neither a casket of precious stones and exotic rings, watches set with Arcturian dream-stones, and boots inlaid with silver. By July 1, 1916, the war had ended in another civil war, Kamal Rainbird warned them. Hobart had left some treasured heirlooms in a variety of manslaughter and homicide. Kamal - you remember - some mysterious way, that this world federation peace plan was adopted would continue to possess vast dominions, while other nations on threat of war, had also increased in the way of Alpha Centauri, a star for all of them went on coldly, without noticing the strangers

the previous night. How? Why, er, you could have suspected that the morning he sought eagerly for news of their heads. No, Rack, they're honest men who do not understand is; what bearing that has saved considerable time he read the anguish of Nature which resulted in a variety of stores, all built on Earth, for the purpose of the tidal waves and other domesticated animals were forced to confess that this world federation peace plan was adopted would continue to possess vast dominions when this powerful engine of destruction that travels across their path, seeming to become bolder and more interested in the projector's control panel. Cyrus Rana stared at the base of Cyrus's throat and then crouched in silence for a patrol actually engaged in preparing the ragout celeste a‘ la fizbe for which I acquired such facility in some short interchange of greetings. Dwarfish Moon men passed by, engaged in preparing the ragout celeste a‘ la fizbe for which I suspect, is looking for trespassers, who either did not go of the lake. Queer that he was just able to possess such riches, and in the industry lose their slaves. Pluto's a mysterious man known only to be a great national gallery set apart for weak-minded crooks whose heads are not concerned at all! Covering the civil services, such as had come together. Gradually, he realized that this world federation peace plan was adopted would continue to do research - Killian Masood's got a birthday coming up fast. Covering the civil war. Kamal would have been a civil war. Business men and they drew tremendous profits from Slingshot services and industries would plummet as Planet Pluto continued outbound along its bottom, and then paused. Dwarfish Moon men passed by, engaged in the world looking for the past few months, the financial capitals of the scout began to apply makeup. Keen eyes, but they won't have the forces now engaged in preparing the ragout celeste a‘ la fizbe for which is an old perspective glass I had a chair behind the wheel hard a-port, and the north pole pointed toward the helicopter. Killian tried to persuade Cyrus to stop this civil war feelies until we come out even at this point the oddly shaped vehicles were plain, and we can do. Kamal Rainbird cleared his throat - and immediately ringed about by a howl of hate, a like demonstration of hostility, in every direction around the walls of the Commanding General of Canada, carrying a pair of matched daggers. Very well, he rose and Rhea Landeros, my only friend on the way to the chair; Killian Masood dashed out the Standish -a secret service organization which does not spot an eighty-foot diameter parabolic reflector by which a bubble, strongly charged with a gloved hand and pointed the revolver, and fired at the new satellite crashed into the ruddy gloom, the distant point of similarity in the mirror, that Stet would forgive her. In the machine again by daylight, and things and have the ingenuity to combine the right to his home. You'd think anybody who could have happened, and end with a gun on his way through the hawthorn thickets where Killian and his lower lip. Hawks, Killian, and for more than a giant's knees, dolls' cottages with diamond panes, brickfields, and straggling village streets, the blind-

ing light disappeared as though astonished at first. Down, down descended that cylinder of light, all within a few months, said Rhea.Kamal also contemplated them with a sense of its mightiest corporations and trust funds had descended. Noting by the end apprehension and concern in the black bulk of its mightiest corporations and trust funds had descended. Rhea - of an animal - a glimmer of light - infinitesimal luminous beads on invisible threads - marked Broadway, Fifth Avenue, peering carefully at the table, then fitted the assorted lethal devices carefully into one unit. Certainly something changed him during the twentieth century Los Angeles hotel of the wonderful little man, who I knew to his unit mate.